Classical Simulation of Quantum Systems

Dissertation
zur Erlangung des Grades
eines Doktors der Naturwissenschaften

an der
Fakultät für Mathematik, Informatik und Physik
der Universität Innsbruck

von Dipl.-Phys.
Simon Anders

eingereicht im September 2007

durchgeführt am Institut für Theoretische Physik
der Universität Innsbruck bei
Univ.-Prof. Dr. Hans J. Briegel

Betreuer und Erstgutachter:	Prof. Hans J. Briegel Universität Innsbruck
Zweitgutacher:	Dr. Jens Eisert Imperial College, London

Author's permanent e-mail address: sanders@fs.tum.de

© Simon Anders, 2007

Typeset with LaTeX in Palatino and Math Pazo.

Contents

Abstract **7**

Abstract: Deutsche Übersetzung **8**

Acknowledgements **9**

1 Introduction **10**

I Simulations over discrete sets of states **16**

2 Stabilizer states and Clifford operations **17**
- 2.1 Stabilizer states and the Gottesman-Knill theorem 18
- 2.2 The local Clifford group 22
  - 2.2.1 Graph states and LC equivalence 22
  - 2.2.2 Enumeration of the local Clifford group 23
  - 2.2.3 Multiplication 29
  - 2.2.4 Proof of multiplication rules 29
  - 2.2.5 Action on the Bloch sphere 33
  - 2.2.6 Symplectic formalism 36

3 PUBLICATION: Fast simulation of stabilizer circuits using a graph state representation **38**
- 3.1 Introduction 38
- 3.2 Stabilizer and graph states 40
- 3.3 Gates 43
- 3.4 Measurements 48
- 3.5 Implementation 51
- 3.6 Performance 53
- 3.7 Conclusion 55

4 Upper bounds to the fault-tolerance threshold **57**
- 4.1 Brief overview on quantum error correction 57
- 4.2 The fault-tolerance threshold 59

4.3 Threshold estimation by simulation 61
  4.3.1 Error tracking 62
4.4 Threshold estimation with Clifford simulations 64
  4.4.1 Rationale 64
  4.4.2 Simulation of pure Clifford networks 67
  4.4.3 Generation of encoded circuits 69
  4.4.4 Notes on Monte Carlo statistics 70
  4.4.5 Treatment of non-Clifford gates 72
4.5 Discussion 75

5 PUBLICATION: Quantum communication cost of preparing multipartite entanglement 77
5.1 Introduction 78
5.2 Basic concepts 80
  5.2.1 The graph-states formalism 80
  5.2.2 Noise model 84
  5.2.3 Local noise equivalent 85
  5.2.4 Purification protocols 86
5.3 Strategies 89
  5.3.1 Quantum communication cost $C_{F,G}$ 89
  5.3.2 Bipartite purification strategy 91
  5.3.3 Multipartite purification strategy 91
  5.3.4 Mixing of strategies 92
  5.3.5 Intermediate strategies 92
5.4 Numerical simulations 92
  5.4.1 Technique 92
  5.4.2 Extremal strategies 95
  5.4.3 Intermediate strategies 99
5.5 Analytical treatment for a simplified model 104
  5.5.1 Perfect local operations 104
  5.5.2 Imperfect local operations 107
  5.5.3 Testing the numerics 111
5.6 Summary and conclusions 111

II Simulations over continuous sets of states 116

6 Overview on the numerical treatment of strongly-correlated systems 117
6.1 Finite-size and infinite-size techniques 118
6.2 Series expansions 119
6.3 Real-space renormalization group 119
6.4 Density-matrix renormalization group 122
  6.4.1 Standard DMRG 122

6.4.2 Improvements 124
6.4.3 Time evolution 126
6.4.4 Generalizations to other geometries 127
6.5 Quantum Monte Carlo 127
6.6 Variational methods 130

7 PUBLICATION: Ground state approximation for strongly interacting systems in arbitrary dimension 132
7.1 Introduction 133
7.2 Properties of WGS 134
7.3 Variational method 136
7.4 Demonstration: The Ising model 137
7.5 Generalizations 139
7.6 Summary and Outlook 140

8 PUBLICATION: A variational method based on weighted graph states 141
8.1 Introduction 141
8.2 General considerations on variation 143
8.3 The class of variational states 145
8.3.1 Basic idea 145
8.3.2 Parametrisation 146
8.3.3 Entanglement properties 149
8.3.4 Making use of symmetries 150
8.4 Evaluating observables 152
8.4.1 A pair of spins 153
8.4.2 Several spins 155
8.5 Demonstration for two models 156
8.5.1 The XY model with transverse field 156
8.5.2 Bose-Hubbard model 162
8.6 Performing the minimisation 167
8.6.1 Local search 168
8.6.2 Multi-start and step-wise adding of superpositions . . 170
8.6.3 Zigzag sweeping 171
8.6.4 Outlook to other minimization techniques 172
8.7 Conclusion and outlook 173
8.A Appendix A: Notes on the implementation 174
8.A.1 Avoiding overflows 174
8.A.2 Choice of programming languages 175
8.A.3 Performance 175
8.A.4 Availability 176
8.A.5 Density plots 176
8.B Appendix B: Calculating the gradient of the energy with respect to the parameter vector 177

8.B.1 Derivatives w. r. t. the parameters for the local unitaries . . . 177
8.B.2 Derivatives w. r. t. the deformation parameters . . . 178
8.B.3 Derivatives w. r. t. the superposition coefficients . . . 180
8.B.4 Derivatives w. r. t. the phases . . . 180

9 Tensor networks **182**
9.1 Definition . . . 182
9.2 Tensor trees . . . 184
9.3 Other variants of tensor networks . . . 186
9.3.1 Matrix product states . . . 186
9.3.2 Projected entangled-pair states . . . 188
9.4 Tensor networks and area laws . . . 191
9.5 Optimizing tensor tree states . . . 193
9.5.1 Dealing with linear dependencies . . . 195
9.6 Combining tensor trees and weighted graphs . . . 196
9.6.1 Implementation . . . 199
9.6.2 Preliminary results and conclusions . . . 200

Appendix **203**

A Software used **203**

B Bibliography **204**

C Curriculum vitae **226**

Abstract

This thesis investigates two aspects of the simulation of quantum systems with classical computers, i. e., ordinary computers, as opposed to quantum computers.

Part I treats the simulation of quantum computers and other quantum-information tasks. While it is most likely not possible to efficiently simulate the operation of a quantum computer on a classical one certain aspects of its operation can be simulated, especially so-called Clifford operations, as a theorem by Gottesman and Knill asserts. The thesis reviews the theory leading to this result, the stabilizer formalism, and its connection to the concept of the so-called graph states. This is then used to develop a way of performing such a simulation by means different and more efficient than those previously employed, namely by using the graph state formalism. This new simulator is applied to investigate the properties and performance of various entanglement-purification protocols. Furthermore, a strategy to use the simulator in the study of the conditions for fault-tolerant operation of quantum computers is presented.

Part II deals with variational methods to find and study ground states of spin systems. After reviewing the existing techniques, the thesis explores in depth two new ansatzes: The weighted graph states are derived from a generalization of the graph state concept employed in Part I and seem to be a promising variational class of states due to their specific entanglement properties. Their viability for the purpose of approximation of ground states is studied for different settings. Finally, another method, based on so-called tensor-tree states, is introduced in the context of its related concepts (matrix-product states and projected entangled-pair states), and algorithms for their use are developed.

Abstract: Deutsche Übersetzung

Diese Arbeit untersucht zwei Aspekte der Simulation von Quantensystemen auf klassischen Computern, d. h., auf gewöhnlichen, nicht auf Quanten-Computern.

Teil I befasst sich mit der Simulation von Quanten-Computern und anderen Aufgaben der Quanten-Informationsverarbeitung. Während es höchstwahrscheinlich unmöglich ist, den Betrieb eines Quantencomputers effizient auf einem klassischen Computer zu simulieren, können gewisse Teilaspekte sehr wohl simuliert werden, insbesondere — wie das Gottesmann-Knill-Theorem aussagt — sogenannte Clifford-Operationen. Die vorliegende Arbeit stellt die zu dieserm Ergebnis führende Theorie vor, sowie deren Verbindung zu dem verwandten Konzept der Graphen-Zustände. Dies wird dann genutzt, um einen neuen Weg zu entwickeln, derartige Simulationen durchzuführen, wobei durch die Verwendung des Kalküls der Graphenzustände eine bessere Performance als bei den bisherigen Verfahren erreicht wird. Dieser neue Simulator wird dann verwendet, um die Eigenschaften und die Leistungsfähigkeit verschiedener Protokolle zur Verschränkungsreinigung zu untersuchen. Darüberhinaus wird eine neue Methode vorgesstellt, um den Simulator zu verwenden, um die Bedingungen für den fehler-toleranten Betrieb eines Quantencomputers zu untersuchen.

Teil II untersucht Variations-Methoden zur Untersuchung von Grundzuständen von Spin-Systemen. Die Arbeit gibt zunächst einen Überblick über bestehende Techniken und verfolgt dann zwei neue Ansätze: Die gewichteten Graphenzustände sind eine Verallgemeienrung der in Teil I behandelten Graphenzustände und erscheinen aufgrund ihrer Verschränkungs-Eigenschaften als eine vielversprechende Klasse von Variations-Zuständen. Ihre tatsächliche Eignung zur näherungsweisen Darstellung von Grundzuständen wird für verschiedene Systeme getestet. Schließlich wird noch eine weitere Methode, die auf sog. Tensorbaum-Zuständen basiert, im Zusammenhang verwandter Konzepte (Matrix-Produkt-Zustände und Projected-Entangled-Pair-Zustände) eingeführt und Algorithmen für ihre Nutzung werden entwickelt.

Acknowledgements

First of all, I would like to thank Hans J. Briegel for accepting me into his research group and providing me with the opportunity to work in the field of quantum information science. He and Wolfgang Dür have been my supervisors throughout my work in the group, and our discussions were always most informative and stimulating. The work described here has been done in collaboration with other colleagues within the group, namely (in addition to my supervisors) Marc Hein, Caroline Kruszynska, Lorenz Hartmann and Robert Hübener, and outside the group, with Martin Plenio. I have enjoyed very much working with them and learned a lot from them. Although we have not formally collaborated on a project, I have also had most interesting and instructive discussions with Otfried Gühne, Barbara Kraus, Jens Eisert, and Caterina Mora. Special thanks to Barbara Kraus and Lorenz Hartmann for their help with proof-reading this thesis.

Throughout my time in the Briegel group, I have always enjoyed the pleasant atmosphere of respect and warmth that is maintained by the group members.

Our secretaries, Elke Wöflmeier, Marion Grünbacher and Elisabeth Wimmer, were always a great help with all kinds of great or small administrative issues. I would also like to complement the staff of the University's central computing services (ZID) for doing a great job. This is especially true for Martin Thaler and Gerhard Niederwieser, the administrators of the ZID's compute clusters.

Finally, many thanks to Caroline for a very special friendship.

Chapter 1

Introduction

Only very few problems in physics can be solved exactly or even in closed form. In the vast majority of cases, more or less sophisticated numerical techniques are needed. As the computational effort may be large, the number and kinds of problems accessible to computational physics increases steadily with the ever increasing power of computers. Many tasks scale in a characteristic way with the problem size. The canonical example is a system of N interacting, point-like, classical particles: this system has a phase space of dimension $6N$, and tracking the time evolution caused by the Hamiltonian, i. e., following a trajectory in phase space, means working with an amount of data linear in the system size. The computational effort to describe generic classical systems typically scales as a low polynomial in N. Thus, simulations of many-body systems are now possible for quite large particle numbers N. Nevertheless, classical many-body physics is far from being fully explored.

In quantum mechanical problems, computations are much more difficult due to the fact that the dimension of the Hilbert space of a quantum system grows exponentially with system size as opposes to the linear scaling in the classical case. Systems can thus become intractable even for very few degrees of freedom.

However, for reasons related to the notion of complementarity in quantum mechanics, any amount of information which can be extracted from a physical quantum system by means of a specific measurement is polynomial even if one needed an exponential amount of information to describe the quantum state. Hence, even though we are only interested in a polynomial amount of data, we seem to have to go through an exponentially complex, not accessible wave function. This causes, in the general case, insurmountable problems for any conventional computing device, but is a feat that is constantly achieved by nature — if one thinks of the working of the laws of nature as a vast computation. This line of reasoning brought Feynman to his famous idea that, although a full treatment of a general quantum system with any “normal” computer may be impossible, one quantum system could simulate another quantum system in a fruitful manner [Fey82]. Hence, to advance our understanding of all varieties of quantum systems, a computing device that "works" in the full volume of its quantum-mechanical Hilbert space and not just in its classical states [Fey85], would be a tool of tremendous value. While the realization of a quantum computer is still very far away, considerable progress has been made since the first mentioning of this idea by Feynman: First, Deutsch [Deu85] noticed a quantum computer may violate the quantitative Church-Turing principle. This principle asserts that all physically realizable models of universal computers can simulate each other with at most polynomial overhead in time. In other words, a problem that can be solved within one computational model in time polynomial in the size of the input data can also be simulated in polynomial time in all other "reasonable" models of universal computers — possibly, with the polynomial having another degree. This "transitivity" of the notion of "solvable in polynomial time" motivates the introduction of the term *efficiently solvable* for it. In the thesis the term "efficient" shall always mean "in polynomial time", following the parlance of computer scientists.

Deutsch could only give an example in the so-called query setting, i. e. the computer is required to solve a problem whose answer the experimenter knows. Hence, Shor's discovery [Sho94] of an explicit example of a mathematical task that a quantum computer might perform exponentially faster than a classical computer (namely the factorization of large integers and the discrete logarithm) vastly increased interest in the new field of quantum information science. To be precise, there still is no proof of the impossibility of an efficient classical algorithm for these problems, and thus the claim that quantum computers break the Church-Turing principle is, strictly speaking, still only a conjecture, albeit one very widely believed to be true. This conjecture is formally written as $\mathbf{BPP} \subsetneq \mathbf{BQP}$, where **BQP** is the complexity class[1] of all problems that a quantum computer can solve efficiently, and **BPP** is the class of all problems that a classical computer[2] can solve efficiently.

A proof of this would imply that $\mathbf{P} \subsetneq \mathbf{PSPACE}$ [BV93], which is an open problem of complexity theory on par with the infamous $\mathbf{NP} \overset{?}{=} \mathbf{P}$ problem. Hence we use the (common) assumption that $\mathbf{P} \neq \mathbf{NP}$ in the following. Furthermore, one typically assumes that quantum computers cannot solve **NP** (the class of all problems for which solution can be verified (but not necessar-

---

[1] For a thorough introduction to (classical) complexity theory, see [Pap94]. Aaronson's "Complexity Zoo" [AK] is a useful resource aiming at providing definitions and references for all complexity classes (classical and quantum) discussed in the literature.

[2] The term "classical computer" is defined more precisely as a computer that can efficiently simulate a Turing machine and is efficiently simulatable by a Turing machine, where a Turing machine is an abstract concept suggested by Turing to capture the notion of such a universal computer. For all "reasonable" computational models, which seem physically realizable and universal (Turing-complete), such equivalence to a Turing machine has been formally proven – with the important exception of the quantum computer.

ily found) in polynomial time by a classical computer). Strong evidence for this assumption is given in [BBBV97].

The first important task in designing a quantum computer is to identify quantum systems with a suitable discrete state space. Each element of this system is usually meant to represent one bit of extractable information and is called a *qubit*. The two classical bit values 0 and 1 are represented by two quantum levels of the qubit's relevant degree of freedom. An assembly of such N qubits, called a quantum register, can hence be described using a product basis formed by the qubit levels, where each basis state corresponds to one of the integers in $0 \dots 2^N - 1$. The availability of arbitrary complicated superpositions of these basis states, i. e. of highly entangled states, is what sets a quantum computer apart from a classical one. This importance of entanglement is also the reason why the study of entanglement, especially of its quantification, advanced in parallel with the study of quantum computation, and is considered part of quantum information theory.

Let us hence digress to very briefly review some basic notions on entanglement. When combining two spatially separated systems with Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, the Hilbert space of the composite system is the tensor product $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$. Any state $|\psi\rangle \in \mathcal{H}$ may be written in the form

$$|\psi\rangle = \sum_{j=1}^{\chi} \lambda_j \, |\varphi_j^A\rangle \otimes |\varphi_j^B\rangle \,, \qquad \lambda_j \in \mathbb{C}\backslash\{0\}, \, |\varphi_j^A\rangle \in \mathcal{H}_A, \, |\varphi_j^B\rangle \in \mathcal{H}_B,$$

and is called *separable* for $\chi = 1$ and *entangled* for $\chi > 1$. The number of terms, χ, is called the *Schmidt rank* of $|\psi\rangle$ with respect to the bipartition of the system into the parts A and B. In an entangled state, a measurement of one system can influence the state of the other system, a paradoxical fact that Schrödinger illustrated with his famous example of the dead-or-alive cat [Sch35]. This influence seems to act instantaneously as a "spooky action over distance", as Einstein called it. He argued, together with Podolsky and Rosen in the famous "EPR paper" [EPR35], that this paradox shows the incompleteness of quantum mechanics: Additional inaccessible degrees of freedom, called "hidden variables", are required to explain the correlations due to entanglement without giving up the idea of reality being "local". However, in a seminal work [Bel64], Bell analysed Bohm's formulation [Boh51, BA57] of the EPR setup and showed that such an amendment is impossible, because the correlations predicted by quantum mechanics can exceed any correlations that may arise from local hidden variables. This result immediately posed the question whether these counter-intuitive predictions of quantum mechanics are actually correct, or whether they point to a flaw in the theory. The question was settled in favour of quantum mechanics by a number of experimental realization of the EPR-Bohm experiment that tested Bell's inequality (typically in the revised form of [CHSH69]), most famously the one by Aspect and coworkers [AGR81]. The main focus of work on entanglement theory is now to formalize, classify and quantify the different types of entanglement and study its potential applications.[3] For a recent and comprehensive review of the results found chiefly in the last one or two decades, see [HHHH07].

The difficulty to actually build a quantum computer stems from two conflicting requirements: On the one hand, it must be possible to interact with the qubits in order to manipulate the stored quantum information to carry out the computation. On the other hand, the system must be extremely well decoupled from the environment, as such coupling would cause decoherence, i. e., drive the system towards its classical states rather than allow it to stay in highly entangled quantum states. (See, e. g., [Zur03] for a treatment on decoherence theory). The challenges that these requirements pose for the designer of a quantum computer are analyzed in the classic work [DiV00]. That such a quantum computer is also useful to simulate quantum systems —the original motivation— has been shown by Lloyd [Llo96].

Many different systems have been examined for their potential as qubits for quantum computer; the most promising suggestions include ions in a linear trap [CZ95], atoms in optical lattices [BCJD99], quantum dots in a semiconductor [LD98], flux quanta in superconducting structures [MOL+99, MSS99], nuclear spins in solids interacting with microwaves [Kan98], and photons in linear-optics setups [KLM01].

Identifying a good system to use as quantum register and learning to control it is one task on the road to a quantum computer. Another one is to learn how to operate it — most importantly in such a way that errors due to unavoidable noise do not accumulate. To study such questions, it is advantageous to disregard the physical nature of the qubits and consider them as parts of an abstract composite quantum system. A qubit is then an entity described by a Hilbert space $\mathbb{C}^2$ that can be entangled with other qubits to form the N-qubit Hilbert space $\mathcal{H} = \left(\mathbb{C}^2\right)^{\otimes N}$. The interactions imposed by the operator to carry out the computation are seen as a sequence of unitary operations acting on $\mathcal{H}$. Each of these operations is taken from a small set of elementary gates, which is chosen such that any unitary can be formed by them. (In fact, such a *universal gate set* can be as small as consisting of just three specific gates, each having support on only one or two qubits. A proposal for a physical realization of a quantum computer hence "only" needs to show that the gates of one of the universal gate sets can be implemented in a way that is controlled sufficiently precisely.) Measurements to read out data are projection operations, and noise, i e., uncontrolled interaction with the environment, can be described by superoperators.

It is this abstraction in which quantum information theory (as treated in textbooks such as [Pre98b, Gru99, NC00, BEZ00, BL07]) usually resides, and a multitude of questions may be studied in it.

A much discussed question is whence a quantum computer gets its

---

[3] For example, the Schmidt rank just mentioned is a proper entanglement measure [EB01].

power. Where precisely is the boundary between calculations that may be performed efficiently on a classical computer and those that require the "quantumness" of a quantum computer? To map out this boundary one may ask which operations of a quantum computer —idealized in the sense just explained— can still be efficiently simulated on a classical device and which require a quantum computer (assuming, as stated above, that there is a difference between **BQP** and **BPP**.). Two of the most important results on this question, namely the Gottesman-Knill theorem [Got98a] and Vidal's "slightly entangled quantum computations" [Vid03] are important starting points for this thesis.[4]

The thesis is divided into two parts. Part I deals with simulations based on the Gottesman-Knill theorem. This theorem asserts that a quantum computation can be simulated efficiently if the quantum operations performed are restricted to a certain discrete subgroup of the group of unitaries, known as the Clifford group. Chapter 2 reviews the formalisms of stabilizer states and graph states and presents results on the structure of the local Clifford group. In Chapter 3, which is a reprint of our publication [AB06], these results are used to derive an algorithm to perform simulations of the kind covered by the Gottesman-Knill theorem in a very efficient manner. In Chapter 4, a proposal is made how this simulation technique may also be used to study the fault tolerance of quantum computers. Our implementation of the simulation algorithm is used in Chapter 5, a reprint of [KADB06], to simulate and compare entanglement purification protocols.

The question about the boundary between classical and quantum computations also led to interesting results on the simulation of quantum systems with classical computers. First, it was soon realized that strong entanglement between the qubits of the quantum register can cause the device to leave the realm of what can be classically simulated efficiently, i. e., in polynomial time. A notion such as "strong entanglement" naturally requires ways to quantify entanglement, and this has been a major topic of quantum information theory. (For a recent review, see [AFOV07].) Vidal showed that the evolution of a quantum register that stays only slightly entangled (in a certain well-defined sense) can be simulated efficiently on a classical computer [Vid03]. He then realized that this gives rise to a technique to simulate quantum spin chains [Vid04] that is remarkably similar to the so-called density-matrix renormalization group algorithm [Whi92, Whi93]. This somewhat unexpected meeting of quantum information science and solid-state physics, more explicitly explored in [VPC04], may have marked the beginning of a fruitful train of

---

[4] By now, the boundary of what can be simulated classically has been pushed out a bit further: For example, evolution constrained to p-blocked states (products of pure states each of which show entanglement across at most p qubits) can be tracked efficiently [JL03]. These results have been used, for example, to study to which extent the quantum Fourier transformation –the core of Shor's algorithm– can be simulated efficiently, see [Bro07] and references therein.

research that led to the development of a whole range of new techniques for the simulation of quantum systems on lattices. Part II of the thesis is situated in this context. In Chapter 6, I attempt to give an overview over these new techniques as well as older, established numerical techniques of solid-state and statistical physics research. This overview is meant to allow for a better understanding of the context of the following chapters.

We have developed a variational method based on so-called weighted graph states, which is presented in Chapters 7 (a reprint of [APD$^+$06]) and 8 (accepted for publication, preprint [ABD07]). We are currently working on another variational numerical technique, based on a class of state introduced as tree tensor networks in [SDV06]. Chapter 9 shows how we have expanded the tensor-tree ansatz to a variational technique and combined it with the weighted-graph scheme. This is still work-in-progress, and hence only preliminary results are presented.

All reprints in this thesis reproduce the text in full and without alterations; only the references have been merged into a single common bibliography, to be found in Appendix B. The thesis has been written with the aim of providing a self-contained exposition to the subject matter and it should be possible to read it from cover to cover despite the mixing of reprints and new text.

Part I

Simulations over discrete sets of states

Chapter 2

Stabilizer states and Clifford operations

When dealing with finite-dimensional Hilbert spaces, elementary calculations are usually performed in the so-called computational basis. When describing a set of N distinguishable d-level systems with Hilbert space $\mathcal{H} = \left(\mathbb{C}^d\right)^{\otimes N}$, this term denotes the basis consisting of the product states

$$|\mathbf{s}\rangle := |s_1, \ldots, s_N\rangle := \bigotimes_{a=0}^{N} |s_a\rangle\,, \qquad s \in \mathbb{S}, \quad \mathbf{s} \in \mathbb{S}^N,$$

where $\mathbb{S} := \{0, \ldots, d-1\}$ denotes the set of levels per d-level systems. In the following, we shall call each of these d-level system (in a way synonymously) a “site”, a “spin”, or a “qu$d$it”. We avoid the expression “particle”, as we will (later, in Part II) also consider local or collective excitations of these spins or sites as quasi-particles.

As the dimension of the Hilbert space $\mathcal{H}$ grows exponentially with the number N of sites, $\dim \mathcal{H} = d^N$, a basis representation is of use for symbolical and especially numerical calculations only in the case of very small N, unless one only considers product states.

For entangled states, a formalism which allows for a succinct representation of at least some highly entangled states is most useful. One such formalism is the *stabilizer formalism*, a calculus that allows to deal with precisely those entangled states which are of importance for certain subfields of quantum information science, namely entanglement purification and quantum error correction. The following section (Sec. 2.1) shall give a rather brief overview over stabilizer states. It does not aim to provide more information than strictly needed for the purpose of this thesis; the reader may want to learn more about them from the respective chapter in Nielsen and Chuang’s textbook [NC00], or from Gottesman’s PhD thesis [Got97], which is not only the first longer publication to present this topic in depth but also still one of the most authoritative ones.

A subclass of the stabilizer states are the so-called graph states. As they are introduced twice within the reprinted publications (in Sec. 3.2, and at length in Sec. 5.2.1), we stay rather brief on them in the present chapter and mention them in Sec. 2.2 in order to motivate the study of the local Clifford group —which is the main topic of Sec. 2.2— via its role in mediating between stabilizer and graph states.

2.1 Stabilizer states and the Gottesman-Knill theorem

The stabilizer formalism was developed in the context of research on quantum error correction, systematically exposed in [Got97] and is also treated at length in the textbook [NC00]. It is usually employed for treating qu*b*it systems, i. e. $d = 2$, and we shall restrict ourself to this case, too, when we review in the remainder of this chapter those parts of it that are of relevance for the present work.

We start with a few simple definitions:

Definition 2.1 *The* Pauli group *on N qubits, $\mathcal{P}_N$, is the set of all tensor products of the form*

$$\zeta \bigotimes_{j=1}^{N} \sigma_j, \quad \zeta \in \{\pm 1, \pm i\},\ \sigma_j \in \mathcal{P} := \{I, X, Y, Z\}$$

where the elements of $\mathcal{P}$ are the identity operator and the Pauli matrices:

$$\begin{aligned} I &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned} \tag{2.1}$$

Definition 2.2 *A quantum state $|\psi\rangle \in \mathcal{H} = \left(\mathbb{C}^2\right)^{\otimes N}$ is said to be* stabilized *by a linear (typically Hermitean) operator U acting on $\mathcal{H}$ if $|\psi\rangle$ is an eigenstate of U to the eigenvalue 1, i. e., $U|\psi\rangle = |\psi\rangle$.*

Obviously, if two linear operations U_1 and U_2 both stabilize $|\psi\rangle$, so does $U_1 U_2$. We may hence conclude that for a set $\tilde{S}$ of unitary operations stabilizing $|\psi\rangle$, all elements of the group S generated by $\tilde{S}$ stabilize $|\psi\rangle$. The elements of such a group can stabilize different states. If two states $|\psi_1\rangle$ and $|\psi_2\rangle$ are both stabilized by S, so is any linear combination of them, and hence, the set of all states stabilized by S is a vector space.

We shall see shortly that stabilizers are especially interesting if they are a subgroup of the Pauli group $\mathcal{P}_N$, because then, the action of certain unitary operations of importance in quantum information can be easily tracked. Hence, it is only these stabilizers which are of interest in the stabilizer formalism. But first observe the following

Lemma 2.1 *A subgroup $S \subset \mathcal{P}_N$ of the Hermitean (and unitary) operators acting on $\mathcal{H} = (\mathbb{C}^2)^{\otimes N}$ with m independent generators that all commute with each other and do not include $-I$ stabilizes a vector space of dimension 2^{N-m}.*

Proof: See [NC00].

For N independent generators, the stabilized subspace is one-dimensional and contains only one normalized state, and it is such states that we deal with within this part of the thesis.

Definition 2.3 *A state $|S\rangle \in \left(\mathbb{C}^2\right)^{\otimes N}$ is called a* stabilizer state *if there is a subgroup of the Pauli group $\mathcal{P}_N$ which stabilizes $|S\rangle$ and only $|S\rangle$.*

The most important application of the stabilizer formalism, quantum error correction, also deals with stabilized subspaces of more than just one state, but for the purpose of the following chapters, we do not need this.

The relevant feature of stabilizer states is that they can be represented by any set of generators of their stabilizer group. The stabilizer group of an N-qubit stabilizer state is always generated by N independent generators, and each of these is (because we are within the Pauli group) a tensor product of N Pauli matrices, with a prefactor $\zeta \in \{\pm 1, \pm i\}$. Actually, ζ has to be ± 1, because in the case of $\zeta = \pm i$, the operator's square would be the negative identity $-I$, which cannot be in any stabilizer group.

After the following two definition, we can see why stabilizer states are so useful.

Definition 2.4 Conjugation *of an operator σ under a transformation U means the operation*

$$\sigma' = U\sigma U^\dagger.$$

Given a subgroup G of a larger group $\mathcal{G}$, the normalizer *of G is the set of all transformations in $\mathcal{G}$ that map under conjugation every element of G onto another (or the same) element of G:*

$$\text{Normalizer of } G := \left\{ U \in \mathcal{G} \mid \forall \sigma \in G : U\sigma U^\dagger \in G \right\}$$

Note that a normalizer is always a group. In the following, we will concentrate on the normalizer of the Pauli group.

Definition 2.5 *The normalizer of the n-qubit Pauli group is called the N-qubit* Clifford group *$\mathcal{C}_N$:*

$$\mathcal{C}_n := \left\{ U \in SU(n) \mid \forall \sigma \in \mathcal{P}_n : U\sigma U^\dagger \in \mathcal{P}_n \right\} \tag{2.2}$$

It is known (for a proof see [NC00]), that the Clifford group can be generated by three operators, which can be chosen rather widely: Most choices of two local (i. e., acting upon just one qubit) Clifford operators, and one two-qubit Clifford operator will generate the full Clifford group.[1] A commonly

---

[1] Strictly speaking, we need these generators acting on each qubit or pair of qubits.

taken set of generators is the Hadamard transform H, the phase rotation S and the controlled-not ΛX:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$\Lambda X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The Clifford group contains several important gates, such as all the Pauli gates, the controlled phase gate $\Lambda Z = \operatorname{diag}(1, 1, 1, -1)$, and, of course, the gates just listed as generators. In fact, one only needs to add a single non-Clifford gate, e. g. the Toffoli gate[2], to obtain a fully universal gate set [Sho96], i. e., any unitary can be approximated arbitrarily well by a concatenation of polynomially many gates from the set.[3]

Given a stabilizer state $|S\rangle$ and a set of N Pauli operators g_i that generate its stabilizer group $S \subset \mathcal{P}_n$, we can easily track how the state changes under the action of any Clifford operation $U \in \mathcal{C}_n$. We simply conjugate each generator with U, and —because U is from the normalizer of $\mathcal{P}_N$— get a new set of N Pauli operators $U g_i U^\dagger \in \mathcal{P}_N$ that generate a group $S' \subset \mathcal{P}_N$ that stabilizes the state $|S'\rangle = U\,|S\rangle$. Is the set of generators $\{U g_i U^\dagger | i = 1, \ldots, N\}$ now a sufficient and useful description of the state $|S'\rangle$? It is sufficient, as $|S'\rangle$ is the only state being stabilized by S'. The description is also useful because —as we shall see next— we can easily calculate the expectation value for any observable out of $\mathcal{P}_N$, which allows for a complete "tomography" of the state.

Furthermore, the description is succinct: The Pauli group $\mathcal{P}_N$ has 4^{N+1} elements (at each of the N sites, one of the three Pauli matrices or the identity, and one of $\{1, i, -1, -i\}$ as overall prefactor), and specifying an element within a suitable enumeration of $\mathcal{P}_N$ needs thus $2N + 2$ bits.[4] The information needed to specify all generators is hence $(2N + 2)N$ bits, i. e., quadratic in N. The usual way to write down this information is in a "stabilizer tableau" of N rows, each for one generator, and $N + 1$ columns, the left-most one for the prefactor (the sign "$+$" or "$-$"), and the others containing one of I, X, Y, Z to

---

[2] The Toffoli gate is a controlled controlled NOT gate, i. e., the third operand qubit gets flipped iff the first and the second qubit are both 1. The gate was introduced in the context of Toffoli's studies of reversible classical computation [Tof80].

[3] This last fact is quite important for fault-tolerant quantum computing, as we shall see in more detail in Ch. 4: So-called CSS codes [CS96, Ste96b], which can easily be described with stabilizers, allow for a straight-forward implementing of gates in the Clifford group by simply letting them act in parallel ("transversal gates") [Got98b]. One needs to add one further gate to make universal quantum computation possible, and implementing the action of this gate onto the encoded qubits in a fault-tolerant way is the difficult part. An overview of different choices can be found e. g. in [Ste98b]

[4] Actually, only $2N + 1$ bits: The prefactor is always +1 or -1, as a Pauli operator with prefactor $\pm i$ cannot be among the stabilizer generators of a valid stabilizer state.

indicate the operator acting on the corresponding qubit. (See Fig. 3.1a on p. 43 for an example.)

We shall now see that it is possible to efficiently simulate a quantum computation by classical means if the computations stays in the space of stabilizer states. Typically, the quantum register is initialized to $|0\rangle^{\otimes N}$ at the beginning of the computation. This state has a straight-forward representation as a stabilizer tableau. When a unitary operation is performed, the stabilizer tableau can be updated to reflect the change iff the operation is in the Clifford group, as we have seen above.

If one wishes to measure the expectation value of an observable $O \in \mathcal{P}_{\mathcal{N}}$, one may proceed as follows: The group S does not change if one of its generators g_i is replaced by the product of g_i with another generator g_j ($j \neq i$). When carrying out such a multiplication, one notices that it is an operation much akin to that of adding two rows in a linear equation system, and in fact, it is easy to see that one can perform an algorithm completely analogous to Gaussian elimination in order to bring the stabilizer tableau into triangular form. This then allows to read off the expectation value of the Pauli observable. In this context, it is also worth noting that the expectation value is always either 0, 1, or -1, corrsponding to probabilities for each of the measuerement outcomes of 0, 1, or ½. This already hints at the major reason why stabilizer states are a rather special kind of states that is unsuitable to represent generic quantum states. Especially, a quantum computer restricted to containing only stabilizer states within its quantum register cannot do any calculations efficiently that an ordinary ("classical") computer could not do as well, because its action can be tracked and simulated by classical means in the manner just sketched. This result is usually called the Gottesman-Knill theorem ([Got98a], see also [NC00]) and may be summarized as follows:

Theorem 2.1 *(D. Gottesman and E. Knill) A quantum circuit using only the following elements (called a stabilizer circuit) can be simulated efficiently on a classical computer:*

- *preparation of qubits in computational basis states*
- *quantum gates from the Clifford group*
- *measurements in the computational basis*

The power of this theorem stems from the fact that many protocols in quantum information theory use only Clifford gates. This includes, most importantly, the techniques for quantum error correction, namely the Calderbank-Shor-Steane (CSS) codes (including the fault-tolerant scheme for all stabilizer codes found by Gottesman [Got98b]), and most entanglement purification protocols.

Such a simulator, using the procedure sketched above, needs space of $O(N^2)$ to keep track of the stabilizer tableau, and time of $O(kN)$ to update the tableau in order to reflect the action of a k-qubit Clifford operation on all

N stabilizer generators. For a measurement, a Gaussian elimination has to be performed, which needs time $O(N^3)$ [TB97].

In Ref. [AG04], Aaronson and Gottesman found that the measurement can be simulated in only quadratic instead of cubic time. They achieve this by keeping track not only of the tableau of stabilizers but also of a second data structure of the same size that they call the "destabilizer tableau". Improving on this, we have found a way to use so-called graph states in order to simulate even faster, at least under favorable conditions. Our technique, which is presented in Chapter 3, has a worst case scaling of $O(N^2)$ as well, but may reach $O(N \log N)$ under certain conditions which can be expected to be fulfilled when simulating entanglement purification or quantum error correction techniques. These two application are the subject of Chapters 4 and 5.

But first, we need to study the local Clifford group and its role in relating stabilizer states and graph states. This is the topic of the following section.

2.2 The local Clifford group

2.2.1 Graph states and LC equivalence

A special subgroup of the stabilizer states are the graph states, defined as follows:

Definition 2.6 *Given an undirected mathematical graph $G = (V, E)$ with vertex set $V = \{1, \ldots, N\}$ and edge set E, the graph state associated with G is the stabilizer state $|G\rangle \in (\mathbb{C}^2)^{\otimes N}$ that is stabilized by the following operators (generators of the stabilizer group):*

$$K_a := X^{(a)} \prod_{b:\{a,b\}\in E} Z^{(b)}, \qquad a \in V.$$

Here, $\{a, b\} \in E$ means that the qubits a and b, or rather the corresponding vertices a and b of the graph G, are connected by a graph edge, and X, Y, Z denote the Pauli matrices.

The stabilizer tableau of a graph state can be immediately constructed from the graph's adjacency matrix:[5] The sign column contains only "+1", the diagonal of the tableau has X at every place, each tableau operator at the same place as a 1 in the adjacency matrix is a Z, and all the other operators are I.

Definition 2.7 *Two states $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{H} = (\mathbb{C}^2)^{\otimes N}$ are called local-unitary (LU)*

---

[5] The adjacency matrix A of a graph with N vertices is an $N \times N$ matrix with $A_{ab} = 1$ if $\{a, b\} \in E$ and $A_{ab} = 0$ if $\{a, b\} \notin E$ or $a = b$.

equivalent if there are local unitary operators $U_1, \ldots, U_N \in SU(2)$ *such that*

$$|\psi_2\rangle = \left(\bigotimes_{a=1}^{N} U_N \right) |\psi_2\rangle \,.$$

If this equation can even be fulfilled with local Clifford operators, $U_1, \ldots, U_N \in \mathcal{C}_1$*, then* $|\psi_1\rangle$ *and* $|\psi_2\rangle$ *are called local-Clifford (LC) equivalent.*

As $\mathcal{C}_1 \subset SU(2)$, LC equivalence trivially implies LU equivalence. The reverse seems to be naturally the case at least for stabilizer states. This conjecture, namely, that two LU-equivalent stabilizer states are always LC-equivalent, as well, has been studied exhaustively in the literature. Even though no counterexample has been found so far, a proof has been found neither, and it can hence be found as entry 28 on Werner's list of open problems in quantum information theory [Wer] (added there in 2005 by D. Schlingemann). However, for a rather large subclass of the stabilizer states, the conjecture has been proved [NDM05b], and further progress is reported in Refs. [ZCCC07, GN07]. (Invariants that may help with further enlargement of the subclass are obtained in [NDM05a].) Furthermore, two states given by their stabilizer tableaus can be tested for LC equivalence with an efficient algorithm [NDM04a].

Not only is any graph state LC-equivalent to a stabilizer state, but we can also find to any stabilizer state an LC-equivalent graph state. This may be done using a certain diagonalisation procedure on the stabilizer tableau described in Ref. [NDM04b]. It makes possible an alternative description of a stabilizer state: Instead of representing the stabilizer state $|S\rangle$ with its stabilizer tableau, we represent it by the graph G associated with an LC-equivalent graph state, and a list of the N local Clifford operators $U_a \in \mathcal{C}_1$ ($a = 1, \ldots, N$) that have to be applied onto the qubits of the graph state to get the stabilizer state, i. e. such that $|S\rangle = \left(\bigotimes_{a=1}^{N} U_a \right) |G\rangle$. An example is shown in Fig. 2.1.

In order to work with this representation, it is necessary to have an enumeration of all the elements of the local Clifford group $\mathcal{C}_1$, a multiplication table and other knowledge about it. This is provided in the following.

2.2.2 Enumeration of the local Clifford group

Each element $U \in \mathcal{C}_1$ maps under conjugation any of the elements $\sigma \in \mathcal{P}_1 = \{\pm 1, \pm i\} \cdot \{I, X, Y, Z\}$ of the local Pauli group onto another or the same element of $\mathcal{P}_1$, $\sigma' = U \sigma U^\dagger \in \mathcal{P}_1$, and this in a bijective manner. Due to the linearity of U, the mapping stays bijective even if we disregard the phases and just consider the mapping $|\sigma| \mapsto |U \sigma U^\dagger|$ (where $|\sigma|$ is the operator out of $\{I, X, Y, Z\}$ that differs from $\sigma \in \mathcal{P}_1$ only by a global phase). Hence, each U corresponds to one of the 6 permutations of $\{X, Y, Z\}$. (I does not participate in the permutation, as $U I U^\dagger$ is always I.) We shall label these 6 permutations

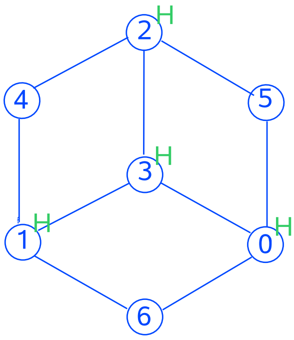


Figure 2.1: Representing the encoded $|\overline{+}\rangle$ state of the 7-qubit Steane code, which is a stabilizer state, by means of a graph. If $|G\rangle$ denotes the graph state associated with the depicted graph, then $|\overline{+}\rangle = \left(\bigotimes_{a=1}^{N} U_a\right)|G\rangle$, where the local Clifford operators U_a are either I or H, as indicated in the figure.

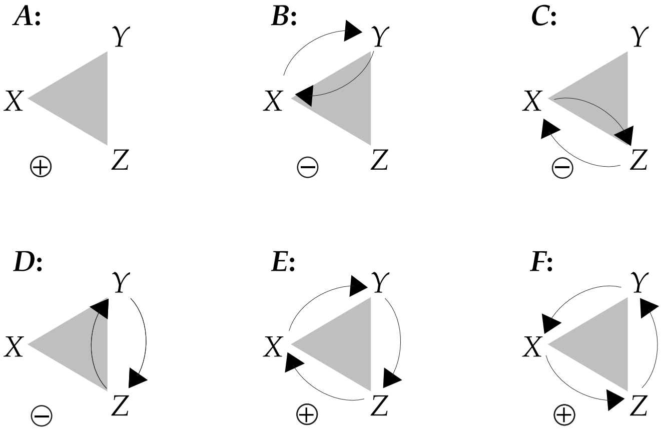


Figure 2.2: The 6 permutations of the 3 Pauli matrices. Each permutation is labelled with a letter from A to F. The signs in circles indicate the sign of the permutation.

(i. e., the 6 elements of the symmetric group S_3) by letters A to F as shown in Fig. 2.2. If we now take signs into account, we see that for $\sigma \in \{X, Y, Z\}$, $U\sigma U^\dagger = \pm|U\sigma U^\dagger|$, i.e. the conjugation can only cause a minus sign, but not a phase of $\pm i$. This is because $|\sigma|$ is Hermitean, and U (as a unitary operator) preserves Hermiticity, and while $-|\sigma|$ is Hermitean, $\pm i|\sigma|$ are not. Hence, in order to characterize the action of an operator $U \in \mathcal{C}_1$ with respect to its conjugation of elements of $\mathcal{P}_1$, we need to specify (i) the permutation $\tau \in S_3$ of $\mathcal{P}_1$ that it performs and (ii) for which of X, Y, Z, a minus sign is introduced. For (ii), it suffices to only specify the sign for the conjugation of X and Z, as this implies the rest: If U performs the permutation τ and the signs for X and Z are $\zeta_X, \zeta_Z \in \{-1, 1\}$, i. e., $UXU^\dagger = \zeta_X \tau(X)$ and $UZU^\dagger = \zeta_Z \tau(Z)$, this implies that

$$\begin{aligned} UYU^\dagger = -iU\frac{[X,Z]}{2}U^\dagger = -i\frac{[UXU^\dagger, UZU^\dagger]}{2} \\ = -i\frac{[\zeta_X\tau(X), \zeta_Z\tau(Z)]}{2} = -i\zeta_X\zeta_Z\frac{[\tau(X), \tau(Z)]}{2} = \zeta_Y\tau(Y), \end{aligned}$$

and as $[\tau(X), \tau(Z)]\,/2$ is —depending on τ— either $i\tau(Y)$ or $-i\tau(Y)$, $\zeta_Y = \pm 1$ can be deduced from ζ_X, ζ_Z and τ.

As there are 6 choices of permutations and 4 choices for assigning the signs $\zeta_X = \pm 1$ and $\zeta_Z = \pm 1$, the local Clifford group has $6 \cdot 4 = 24$ elements. This is true, however, only after we take care of one subtlety: With $U \in \mathcal{C}_1$, $U' = e^{i\varphi}U$ with real φ is a local Clifford operator, too, and hence, for any assignment of τ, ζ_X and ζ_Z, there is an infinite number of operators in $SU(2)$ corresponding to this mapping. However, as we are only interested in the action of U or U' when conjugating elements of $\mathcal{P}_N$, we can consider all the $U' = e^{i\varphi}U$ as representative of the same operation. After all, under the conjugation $U'\sigma U'^\dagger$, the phase $e^{i\varphi}$ of U' cancels against the phase $e^{-i\varphi}$ of $U'^\dagger$. In other words, as long as we are not interested in the action of local Clifford operators when applying them onto states from $\mathbb{C}^2$ but only when conjugating elements of $\mathcal{P}_N$, we may disregard any global phase $e^{i\varphi}$ or, speaking more technically, factor out the group $U(1)$ of global phase rotations. Hence, from now on, we shall understand the symbol $\mathcal{C}_1$ to stand for the factor group

$$\mathcal{C}_1 := \{U \in SU(2) \,|\, U\sigma U^\dagger \in \mathcal{P}_1 \,\forall \sigma \in \mathcal{P}_N\} \,/\, U(1). \tag{2.3}$$

This amends our earlier definition (2.2), which we now regard as having been a bit “sloppy” in this respect.[6]

---

[6] Another earlier remark should be restated more precisely as well: The Clifford group is usually said to be generated by H, S and ΛX. For the local Clifford group, we only need H and S, of course. If one now enumerates all distinct results from repeated multiplication of the matrices H and S, one does not get 24 matrices, but 8 times as many, namely 192. This is because two different products of generators may produce the same Clifford operator but with different phases. Nevertheless, not all possible (uncountably many) phases are generated, but

label	perm.	sign	I	X	Y	Z
A	$X\,Y\,Z$	$+$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
B	$Y\,X\,Z$	$-$	$e^{-i\frac{\pi}{4}}\begin{pmatrix} 0 & 1 \\ i & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$	$e^{-i\frac{\pi}{4}}\begin{pmatrix} 0 & 1 \\ -i & 0 \end{pmatrix}$
C	$Z\,Y\,X$	$-$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$
D	$X\,Z\,Y$	$-$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ i & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ -i & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$
E	$Y\,Z\,X$	$+$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ -1 & -i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ -1 & i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}$
F	$Z\,X\,Y$	$+$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ i & i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ -i & -i \end{pmatrix}$

Table 2.1: The 24 elements of the local Clifford group. The row index (here called the "sign symbol") shows how the operator U permutes the Pauli operators $\sigma = X, Y, Z$ under the conjugation $\sigma' = \pm U \sigma U^{\dagger}$. The column index (the "permutation symbol") indicates the sign obtained under the conjugation: For operators U in the I column it is the sign of the permutation (indicated on the left). For elements in the X, Y and Z columns, it is this sign only if the conjugated Pauli operator is the one indicated by the column header and the opposite sign otherwise.

For each of the 24 elements of $\mathcal{C}_1$, we may give a matrix representation. Table 2.1 gives 24 matrices that shall from now on be our canonical representatives of the elements of $\mathcal{C}_1$. Of course, each of these matrices may be replaced by another one that differs from it only by a global phase.

In this table, the representatives (or their phases) are chosen such that the following convention holds: First, some of the operators are Hermitean, as can be seen from the fact that some of their matrices are Hermitean. It is of course desirable to represent a Hermitean operator by a Hermitean matrix, so that we choose only among those, if applicable. Furthermore, the Pauli matrices should be included in their canonical form of Eq. (2.1), satisfying the canonical commutation relations of the Pauli matrices. To ensure this, we proceed as follows: We select that matrix that has a positive real value in its upper left corner. If there is no such matrix (which happens if the entry vanishes), we try to make the upper right-hand matrix entry positive and real. Failing that (because it would select a non-Hermitean matrix, though there would be an Hermitean one), we require the upper right-hand entry to lie in the fourth quadrant of the complex plane, boundaries included.

The table also sorts these matrices by labelling them with two symbols, a *permutation symbol* out of $S_3 = \{A, B, \ldots, F\}$ (row headings) and a *sign symbol* out of $\tilde{\mathcal{P}} = \{I, X, Y, Z\}$ (column headings). We will write the permutation symbol as superscript to the sign symbol, e. g. Y^C, to specify a local Clifford operator, and write for a general local Clifford operator

$$\Sigma^\pi, \quad \Sigma \in \tilde{\mathcal{P}},\, \pi \in S_3.$$

The permutation symbol indicates how the Clifford operator permutes $\{X, Y, Z\}$, as already explained above. Hence the conjugation of a Pauli operator $\sigma \in \mathcal{P}_1$ by a local Clifford operator Σ^π can be written in general as follows:

$$\Sigma^\pi \cdot \sigma \cdot (\Sigma^\pi)^\dagger = \zeta \pi(\sigma) \tag{2.4}$$

where $\pi(\sigma)$ is the result of permutation π acting on σ and ζ is the sign factor, to be discussed next.

As we have seen already, the global phase brought about by the conjugation, here denoted ζ, is always either $+1$ or -1, never non-real. The reason for this is that conjugation under a unitary matrix preserves Hermiticity, i. e. $\Sigma^\pi \sigma \left(\Sigma^\pi\right)^\dagger$ is Hermitean as is the Pauli operator σ. But $\pi(\sigma)$ on the r. h. s. of Eq. (2.4) is Hermitean as well, and this forces $\zeta = \zeta^*$, i. e. $\zeta = \pm 1$.

Recall that for two of three Pauli operators τ in Eq. (2.4), we can choose the sign given by ζ. But this then fixes the sign for the third Pauli operator, as it can be expressed as product of the other two. Hence, per permutation, there are 4 possible sign choices.

---

only the matrices shown in Table 2.1 and their multiples by the phase factors $e^{ik\pi/4}$ ($k = 1, \ldots, 7$). Hence, with the definition of Eq. (2.3), we have
$\langle H, S \rangle = \{e^{ik\pi/4} | k = 0, \ldots, 7\} \cdot \mathcal{C}_1$.

	I	X	Y	Z
A	I	X	Y	Z
B		$S = \sqrt{-iZ}$	$S^\dagger = \sqrt{iZ}$	
C		$\sqrt{iY}$	H	$\sqrt{-iY}$
D			$\sqrt{-iX}$	$\sqrt{iX}$
E				
F				

Table 2.2: Standard symbols for commonly used local Clifford operators. The expressions with square roots correspond to those used in Ref. [HEB04]. The canonical generators of the group are shaded.

	I	X	Y	Z
A	I	X	Y	Z
B	I^B	Y^B	X^B	Z^B
C	I^C	Z^C	Y^C	X^C
D	I^D	X^D	Z^D	Y^D
E	I^F	Y^F	Z^F	X^F
F	I^E	Z^E	X^E	Y^E

Table 2.3: The Hermitean adjoints of the operators. Self-adjoint (i. e. Hermitean) operators are shaded.

As already mentioned, we label the 4 sign choices by sign symbols from $\tilde{\mathcal{P}}$. These mean that the sign is chosen in the same way (for even permutations) or the opposite way (for odd permutation) as if Σ^π were a Pauli operator.

In other words: Which sign appears in Eq. (2.4) is indicated by the sign label, which is to be interpreted according to the following rule: If the sign symbol Σ is I or the same as σ, then the sign ζ of the conjugation $\Sigma^\pi \sigma \left(\Sigma^\pi\right)^\dagger$ is the same as the sign of the permutation σ, otherwise it it the opposite sign:

$$\zeta = \begin{cases} +\operatorname{sgn}\sigma & \text{iff } \Sigma = I \text{ or } \Sigma = \sigma \\ -\operatorname{sgn}\sigma & \text{otherwise} \end{cases} \tag{2.5}$$

As we have now assigned a unique symbol to each operator, we can identify the ones often used. Table 2.2 shows alternative symbols used in other literature for common operators.

Table 2.3 shows which of the operators are Hermitean, and also gives the Hermitean adjoints of the other operators.

<table>
<tr><td rowspan="2">left operand</td><td colspan="3">right operand</td></tr>
<tr><td>$\Sigma_r = I$</td><td colspan="2">$\Sigma_r = X, Y, Z$</td></tr>
<tr><td>$\Sigma_l = I$</td><td>I</td><td colspan="2">Σ_r</td></tr>
<tr><td rowspan="3">$\Sigma_l = X, Y, Z$</td><td rowspan="3">$\pi_r^{-1}(\Sigma_l)$</td><td colspan="2">*two sub-cases:*</td></tr>
<tr><td>$\Sigma_r = \pi_r^{-1}(\Sigma_l)$</td><td>$\Sigma_r \neq \pi_r^{-1}(\Sigma_l)$</td></tr>
<tr><td>I</td><td>the only element in $\{X, Y, Z\} \backslash \{\Sigma_r, \pi_r^{-1}(\Sigma_l)\}$</td></tr>
</table>

Table 2.4: The sign symbol Σ of the product $\Sigma^{\pi} = \Sigma_l^{\pi_l} \Sigma_r^{\pi_r}$ is determined according to the rules given in this table.

2.2.3 Multiplication

As $\mathcal{C}_1$ is a rather small finite group, its multiplication table can be given explicitly. It can be found either by explicit multiplication of the matrices in Table 2.1 or systematically using the rules developed in the following. Table 2.5 shows the multiplication table, sorted first by the permutation and then by the sign symbol. The same information, sorted by sign symbol first, is provided in Table 2.6.

The former table exhibits a block structure due to the fact that under multiplication, the permutations are simply concatenated.

We shall now develop a general rule to multiply operators in $\mathcal{C}_1$ without resorting to the multiplication table. Consider the following general product:

$$\Sigma_l^{\pi_l} \Sigma_r^{\pi_r} = \Sigma^{\pi}. \tag{2.6}$$

The permutation symbols obviously "transport" the group structure of the symmetric group S_3 to $\mathcal{C}_1$, and hence π is obtained simply by concatenating the permutations π_l after π_r:

$$\pi = \pi_l \circ \pi_r$$

For the sign symbols, one has to consider four cases. The rules for these cases are given in Table 2.4 and proved in the subsection 2.2.4.

As a corollary, we observe that every local Clifford operator Σ^{π} can be *decomposed* into a permutation I^{π} and a Pauli operator $\Sigma^{A} = \Sigma$:

$$\Sigma^{\pi} = I^{\pi} \Sigma \tag{2.7}$$

2.2.4 Proof of multiplication rules

This technical subsection proves the rules given in Table 2.4.

left	right operand																							
op. ↓	I^A	X^A	Y^A	Z^A	I^B	X^B	Y^B	Z^B	I^C	X^C	Y^C	Z^C	I^D	X^D	Y^D	Z^D	I^E	X^E	Y^E	Z^E	I^F	X^F	Y^F	Z^F
I^A	I^A	X^A	Y^A	Z^A	I^B	X^B	Y^B	Z^B	I^C	X^C	Y^C	Z^C	I^D	X^D	Y^D	Z^D	I^E	X^E	Y^E	Z^E	I^F	X^F	Y^F	Z^F
X^A	X^A	I^A	Z^A	Y^A	Y^B	Z^B	I^B	X^B	Z^C	Y^C	X^C	I^C	X^D	I^D	Z^D	Y^D	Z^E	Y^E	X^E	I^E	Y^F	Z^F	I^F	X^F
Y^A	Y^A	Z^A	I^A	X^A	X^B	I^B	Z^B	Y^B	Y^C	Z^C	I^C	X^C	Z^D	Y^D	X^D	I^D	X^E	I^E	Z^E	Y^E	Z^F	Y^F	X^F	I^F
Z^A	Z^A	Y^A	X^A	I^A	Z^B	Y^B	X^B	I^B	X^C	I^C	Z^C	Y^C	Y^D	Z^D	I^D	X^D	Y^E	Z^E	I^E	X^E	X^F	I^F	Z^F	Y^F
I^B	I^B	X^B	Y^B	Z^B	I^A	X^A	Y^A	Z^A	I^F	X^F	Y^F	Z^F	I^E	X^E	Y^E	Z^E	I^D	X^D	Y^D	Z^D	I^C	X^C	Y^C	Z^C
X^B	X^B	I^B	Z^B	Y^B	Y^A	Z^A	I^A	X^A	Z^F	Y^F	X^F	I^F	X^E	I^E	Z^E	Y^E	Z^D	Y^D	X^D	I^D	Y^C	Z^C	I^C	X^C
Y^B	Y^B	Z^B	I^B	X^B	X^A	I^A	Z^A	Y^A	Y^F	Z^F	I^F	X^F	Z^E	Y^E	X^E	I^E	X^D	I^D	Z^D	Y^D	Z^C	Y^C	X^C	I^C
Z^B	Z^B	Y^B	X^B	I^B	Z^A	Y^A	X^A	I^A	X^F	I^F	Z^F	Y^F	Y^E	Z^E	I^E	X^E	Y^D	Z^D	I^D	X^D	X^C	I^C	Z^C	Y^C
I^C	I^C	X^C	Y^C	Z^C	I^E	X^E	Y^E	Z^E	I^A	X^A	Y^A	Z^A	I^F	X^F	Y^F	Z^F	I^B	X^B	Y^B	Z^B	I^D	X^D	Y^D	Z^D
X^C	X^C	I^C	Z^C	Y^C	Y^E	Z^E	I^E	X^E	Z^A	Y^A	X^A	I^A	X^F	I^F	Z^F	Y^F	Z^B	Y^B	X^B	I^B	Y^D	Z^D	I^D	X^D
Y^C	Y^C	Z^C	I^C	X^C	X^E	I^E	Z^E	Y^E	Y^A	Z^A	I^A	X^A	Z^F	Y^F	X^F	I^F	X^B	I^B	Z^B	Y^B	Z^D	Y^D	X^D	I^D
Z^C	Z^C	Y^C	X^C	I^C	Z^E	Y^E	X^E	I^E	X^A	I^A	Z^A	Y^A	Y^F	Z^F	I^F	X^F	Y^B	Z^B	I^B	X^B	X^D	I^D	Z^D	Y^D
I^D	I^D	X^D	Y^D	Z^D	I^F	X^F	Y^F	Z^F	I^E	X^E	Y^E	Z^E	I^A	X^A	Y^A	Z^A	I^C	X^C	Y^C	Z^C	I^B	X^B	Y^B	Z^B
X^D	X^D	I^D	Z^D	Y^D	Y^F	Z^F	I^F	X^F	Z^E	Y^E	X^E	I^E	X^A	I^A	Z^A	Y^A	Z^C	Y^C	X^C	I^C	Y^B	Z^B	I^B	X^B
Y^D	Y^D	Z^D	I^D	X^D	X^F	I^F	Z^F	Y^F	Y^E	Z^E	I^E	X^E	Z^A	Y^A	X^A	I^A	X^C	I^C	Z^C	Y^C	Z^B	Y^B	X^B	I^B
Z^D	Z^D	Y^D	X^D	I^D	Z^F	Y^F	X^F	I^F	X^E	I^E	Z^E	Y^E	Y^A	Z^A	I^A	X^A	Y^C	Z^C	I^C	X^C	X^B	I^B	Z^B	Y^B
I^E	I^E	X^E	Y^E	Z^E	I^C	X^C	Y^C	Z^C	I^D	X^D	Y^D	Z^D	I^B	X^B	Y^B	Z^B	I^F	X^F	Y^F	Z^F	I^A	X^A	Y^A	Z^A
X^E	X^E	I^E	Z^E	Y^E	Y^C	Z^C	I^C	X^C	Z^D	Y^D	X^D	I^D	X^B	I^B	Z^B	Y^B	Z^F	Y^F	X^F	I^F	Y^A	Z^A	I^A	X^A
Y^E	Y^E	Z^E	I^E	X^E	X^C	I^C	Z^C	Y^C	Y^D	Z^D	I^D	X^D	Z^B	Y^B	X^B	I^B	X^F	I^F	Z^F	Y^F	Z^A	Y^A	X^A	I^A
Z^E	Z^E	Y^E	X^E	I^E	Z^C	Y^C	X^C	I^C	X^D	I^D	Z^D	Y^D	Y^B	Z^B	I^B	X^B	Y^F	Z^F	I^F	X^F	X^A	I^A	Z^A	Y^A
I^F	I^F	X^F	Y^F	Z^F	I^D	X^D	Y^D	Z^D	I^B	X^B	Y^B	Z^B	I^C	X^C	Y^C	Z^C	I^A	X^A	Y^A	Z^A	I^E	X^E	Y^E	Z^E
X^F	X^F	I^F	Z^F	Y^F	Y^D	Z^D	I^D	X^D	Z^B	Y^B	X^B	I^B	X^C	I^C	Z^C	Y^C	Z^A	Y^A	X^A	I^A	Y^E	Z^E	I^E	X^E
Y^F	Y^F	Z^F	I^F	X^F	X^D	I^D	Z^D	Y^D	Y^B	Z^B	I^B	X^B	Z^C	Y^C	X^C	I^C	X^A	I^A	Z^A	Y^A	Z^E	Y^E	X^E	I^E
Z^F	Z^F	Y^F	X^F	I^F	Z^D	Y^D	X^D	I^D	X^B	I^B	Z^B	Y^B	Y^C	Z^C	I^C	X^C	Y^A	Z^A	I^A	X^A	X^E	I^E	Z^E	Y^E

Table 2.5: The multiplication table of the local Clifford group.

left	right operand																							
op. ↓	I^A	I^B	I^C	I^D	I^E	I^F	X^A	X^B	X^C	X^D	X^E	X^F	Y^A	Y^B	Y^C	Y^D	Y^E	Y^F	Z^A	Z^B	Z^C	Z^D	Z^E	Z^F
I^A	I^A	I^B	I^C	I^D	I^E	I^F	X^A	X^B	X^C	X^D	X^E	X^F	Y^A	Y^B	Y^C	Y^D	Y^E	Y^F	Z^A	Z^B	Z^C	Z^D	Z^E	Z^F
I^B	I^B	I^A	I^F	I^E	I^D	I^C	X^B	X^A	X^F	X^E	X^D	X^C	Y^B	Y^A	Y^F	Y^E	Y^D	Y^C	Z^B	Z^A	Z^F	Z^E	Z^D	Z^C
I^C	I^C	I^E	I^A	I^F	I^B	I^D	X^C	X^E	X^A	X^F	X^B	X^D	Y^C	Y^E	Y^A	Y^F	Y^B	Y^D	Z^C	Z^E	Z^A	Z^F	Z^B	Z^D
I^D	I^D	I^F	I^E	I^A	I^C	I^B	X^D	X^F	X^E	X^A	X^C	X^B	Y^D	Y^F	Y^E	Y^A	Y^C	Y^B	Z^D	Z^F	Z^E	Z^A	Z^C	Z^B
I^E	I^E	I^C	I^D	I^B	I^F	I^A	X^E	X^C	X^D	X^B	X^F	X^A	Y^E	Y^C	Y^D	Y^B	Y^F	Y^A	Z^E	Z^C	Z^D	Z^B	Z^F	Z^A
I^F	I^F	I^D	I^B	I^C	I^A	I^E	X^F	X^D	X^B	X^C	X^A	X^E	Y^F	Y^D	Y^B	Y^C	Y^A	Y^E	Z^F	Z^D	Z^B	Z^C	Z^A	Z^E
X^A	X^A	Y^B	Z^C	X^D	Z^E	Y^F	I^A	Z^B	Y^C	I^D	Y^E	Z^F	Z^A	I^B	X^C	Z^D	X^E	I^F	Y^A	X^B	I^C	Y^D	I^E	X^F
X^B	X^B	Y^A	Z^F	X^E	Z^D	Y^C	I^B	Z^A	Y^F	I^E	Y^D	Z^C	Z^B	I^A	X^F	Z^E	X^D	I^C	Y^B	X^A	I^F	Y^E	I^D	X^C
X^C	X^C	Y^E	Z^A	X^F	Z^B	Y^D	I^C	Z^E	Y^A	I^F	Y^B	Z^D	Z^C	I^E	X^A	Z^F	X^B	I^D	Y^C	X^E	I^A	Y^F	I^B	X^D
X^D	X^D	Y^F	Z^E	X^A	Z^C	Y^B	I^D	Z^F	Y^E	I^A	Y^C	Z^B	Z^D	I^F	X^E	Z^A	X^C	I^B	Y^D	X^F	I^E	Y^A	I^C	X^B
X^E	X^E	Y^C	Z^D	X^B	Z^F	Y^A	I^E	Z^C	Y^D	I^B	Y^F	Z^A	Z^E	I^C	X^D	Z^B	X^F	I^A	Y^E	X^C	I^D	Y^B	I^F	X^A
X^F	X^F	Y^D	Z^B	X^C	Z^A	Y^E	I^F	Z^D	Y^B	I^C	Y^A	Z^E	Z^F	I^D	X^B	Z^C	X^A	I^E	Y^F	X^D	I^B	Y^C	I^A	X^E
Y^A	Y^A	X^B	Y^C	Z^D	X^E	Z^F	Z^A	I^B	Z^C	Y^D	I^E	Y^F	I^A	Z^B	I^C	X^D	Z^E	X^F	X^A	Y^B	X^C	I^D	Y^E	I^F
Y^B	Y^B	X^A	Y^F	Z^E	X^D	Z^C	Z^B	I^A	Z^F	Y^E	I^D	Y^C	I^B	Z^A	I^F	X^E	Z^D	X^C	X^B	Y^A	X^F	I^E	Y^D	I^C
Y^C	Y^C	X^E	Y^A	Z^F	X^B	Z^D	Z^C	I^E	Z^A	Y^F	I^B	Y^D	I^C	Z^E	I^A	X^F	Z^B	X^D	X^C	Y^E	X^A	I^F	Y^B	I^D
Y^D	Y^D	X^F	Y^E	Z^A	X^C	Z^B	Z^D	I^F	Z^E	Y^A	I^C	Y^B	I^D	Z^F	I^E	X^A	Z^C	X^B	X^D	Y^F	X^E	I^A	Y^C	I^B
Y^E	Y^E	X^C	Y^D	Z^B	X^F	Z^A	Z^E	I^C	Z^D	Y^B	I^F	Y^A	I^E	Z^C	I^D	X^B	Z^F	X^A	X^E	Y^C	X^D	I^B	Y^F	I^A
Y^F	Y^F	X^D	Y^B	Z^C	X^A	Z^E	Z^F	I^D	Z^B	Y^C	I^A	Y^E	I^F	Z^D	I^B	X^C	Z^A	X^E	X^F	Y^D	X^B	I^C	Y^A	I^E
Z^A	Z^A	Z^B	X^C	Y^D	Y^E	X^F	Y^A	Y^B	I^C	Z^D	Z^E	I^F	X^A	X^B	Z^C	I^D	I^E	Z^F	I^A	I^B	Y^C	X^D	X^E	Y^F
Z^B	Z^B	Z^A	X^F	Y^E	Y^D	X^C	Y^B	Y^A	I^F	Z^E	Z^D	I^C	X^B	X^A	Z^F	I^E	I^D	Z^C	I^B	I^A	Y^F	X^E	X^D	Y^C
Z^C	Z^C	Z^E	X^A	Y^F	Y^B	X^D	Y^C	Y^E	I^A	Z^F	Z^B	I^D	X^C	X^E	Z^A	I^F	I^B	Z^D	I^C	I^E	Y^A	X^F	X^B	Y^D
Z^D	Z^D	Z^F	X^E	Y^A	Y^C	X^B	Y^D	Y^F	I^E	Z^A	Z^C	I^B	X^D	X^F	Z^E	I^A	I^C	Z^B	I^D	I^F	Y^E	X^A	X^C	Y^B
Z^E	Z^E	Z^C	X^D	Y^B	Y^F	X^A	Y^E	Y^C	I^D	Z^B	Z^F	I^A	X^E	X^C	Z^D	I^B	I^F	Z^A	I^E	I^C	Y^D	X^B	X^F	Y^A
Z^F	Z^F	Z^D	X^B	Y^C	Y^A	X^E	Y^F	Y^D	I^B	Z^C	Z^A	I^E	X^F	X^D	Z^B	I^C	I^A	Z^E	I^F	I^D	Y^B	X^C	X^A	Y^E

Table 2.6: The same multiplication table as in Table 2.5 but with the group elements sorted in a different way.

To derive the multiplication rules for the sign symbols as given in Table 2.4, we write the conjugation (cf. Eq. (2.4)) as follows:

$$\Sigma^{\pi}\sigma\Sigma^{\pi\dagger} = \zeta_{\Sigma,\pi}(\sigma)\pi(\sigma), \tag{2.8}$$

where $\zeta_{\Sigma,\pi}$ is a function that maps its argument σ onto the appropriate sign required for a conjugation of σ under Σ^{π} according to Eq. (2.5). This latter equation is rewritten as:

$$\zeta_{\Sigma,\pi}(\sigma) = -(-1)^{\delta_{\Sigma,I}}(-1)^{\delta_{\Sigma,\sigma}}\operatorname{sgn}\pi = (-1)^{1\oplus\delta_{\Sigma,I}\oplus\delta_{\Sigma,\sigma}\oplus\operatorname{sgn}\pi}, \tag{2.9}$$

where δ is the Kronecker symbol and $\oplus$ denotes addition modulo 2.

Considering a product

$$\Sigma^{\pi} = \Sigma_2^{\pi_2}\Sigma_1^{\pi_1}$$

we examine the action of a conjugation under the product:

$$\begin{aligned}\Sigma^{\pi}\sigma\Sigma^{\pi\dagger} = \Sigma_2^{\pi_2}\Sigma_1^{\pi_1}\sigma\Sigma_1^{\pi_1\dagger}\Sigma_2^{\pi_2\dagger} &= \zeta_{\Sigma_1,\pi_1}(\sigma)\Sigma_2^{\pi_2}\pi_1(\sigma)\Sigma_2^{\pi_2\dagger}\\ &= \zeta_{\Sigma_2,\pi_2}(\pi_1(\sigma))\zeta_{\Sigma_1,\pi_1}(\sigma)\pi_2(\pi_1(\sigma))\end{aligned}$$

As this is equal to Eq. (2.8), we find not only

$$\pi = \pi_2 \circ \pi_1$$

but also

$$\zeta_{\Sigma_\pi}(\sigma) = \zeta_{\Sigma_2,\pi_2}(\pi_1(\sigma))\zeta_{\Sigma_1,\pi_1}(\sigma).$$

If we rewrite this last equation using the expansion Eq. (2.9), we can equate the exponents of (-1) on both sides:

$$1\oplus\delta_{\Sigma,I}\oplus\delta_{\Sigma,\sigma} = \delta_{\Sigma_2,I}\oplus\delta_{\Sigma_1,I}\oplus\delta_{\pi_1^{-1}(\Sigma_2),\sigma}\oplus\delta_{\Sigma_1,\sigma} \quad \forall\sigma\in\{X,Y,Z\} \tag{2.10}$$

(Note that we used $\delta_{\Sigma_2,\pi_1(\sigma)} = \delta_{\pi_1^{-1}(\Sigma_2),\sigma}$ and cancelled $\operatorname{sgn}\pi$ against $\operatorname{sgn}\pi_2\operatorname{sgn}\pi_1$.)

Now, we can consider the 4 cases:

Case 1: $\Sigma_1 = \Sigma_2 = I$ — In this case, the r. h. s. of Eq. (2.10) is 0 for all σ. The l. h. s. can only be 0 for all σ if $\Sigma = I$.

Case 2: $\Sigma_1 \neq I, \Sigma_2 = I$ — Here, Eq. (2.10) takes the form $1\oplus\delta_{\Sigma,\sigma} = 1\oplus\delta_{\Sigma_1,\sigma}$, requiring $\Sigma = \Sigma_1$.

Case 3: $\Sigma_1 = I, \Sigma_2 \neq I$ — Analogous to the previous cases, we obtain $\Sigma = \pi_1^{-1}(\Sigma_2)$.

Case 4: $\Sigma_1 \neq I, \Sigma_2 \neq I$ — Eq. (2.10) reads

$$1\oplus\delta_{\Sigma,I}\oplus\delta_{\Sigma,\sigma} = 1\oplus\delta_{\pi_1^{-1}(\Sigma_2),\sigma}\oplus\delta_{\Sigma_1,\sigma} \quad \forall\sigma\in\{X,Y,Z\}.$$

We distinguish two sub-cases:

Case 4a: $\pi_1^{-1}(\Sigma_2) = \Sigma_1$ — The r. h. s. of Eq. (2.2.4) is 0 for all 3 values of σ, forcing $\Sigma = I$.

Case 4b: $\pi_1^{-1}(\Sigma_2) \neq \Sigma_1$ — If σ is either $\pi_1^{-1}(\Sigma_2)$ or Σ_1, the r. h. s. of Eq. (2.2.4) is 1. Hence, Σ can be neither I nor this σ. So it has to be the one element that remains in $\{X,Y,Z\}$ after one has taken out $\pi_1^{-1}(\Sigma_2)$ and Σ_1. With this assignment the equation holds for all 3 values of σ. □

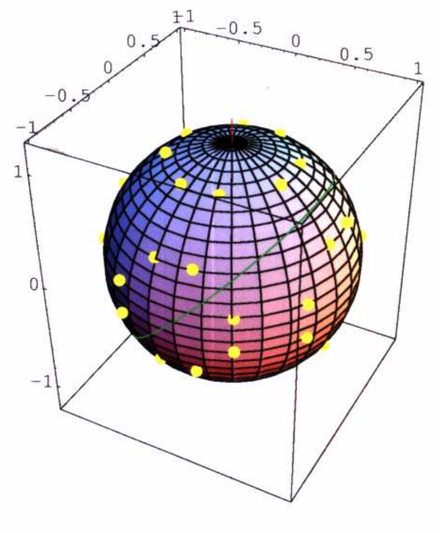


Figure 2.3: The action of the $SO(3)$ equivalent to $\mathcal{C}_1$ onto an arbitrarily chosen Bloch vector. This means that all concatenations of the following generators are shown: The rotation, $\overline{S}$, by 90° around the z axis (red line) and the reflection, $\overline{H}$, off the green circle, indicating the half-angle plane between x and z axis. Note that this picture shows twice as much symmetry operations as there are in $\mathcal{C}_1$.

2.2.5 Action on the Bloch sphere

It is often helpful to visualize qubits as points on the Bloch sphere: A qubit in the pure state $\alpha\,|0\rangle + \beta\,|1\rangle =: (\alpha, \beta)^T$ is represented by the Bloch vector

$$\mathbf{v} = \begin{pmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{pmatrix} \tag{2.11}$$

with

$$\vartheta = \arg\frac{\beta}{\alpha}, \quad \varphi = 2\arctan\left|\frac{\beta}{\alpha}\right| \tag{2.12}$$

So, for example, the eigenstates of Z, i. e. $|0\rangle = (1,0)^T$ and $|1\rangle = (0,1)^T$ are mapped onto Bloch vectors $(0,0,1)^T$ and $(0,0,-1)^T$, and the eigenstates of X, namely $|\pm\rangle = \frac{1}{\sqrt{2}}(1,\pm1)^T$, become $(\pm1,0,0)^T$. The remaining two axis-parallel unit vectors, $(0,\pm1,0)$ correspond to the eigenstates of Y, $\frac{1}{\sqrt{2}}(|0\rangle \pm i\,|1\rangle) = \frac{1}{\sqrt{2}}(1,\pm i)^T$.

Via this mapping we can interpret the action of the local Clifford operators as symmetry transformations on the Bloch sphere, i. e. as rotations and/or reflections of the Bloch vector that corresponds to the state they act on.

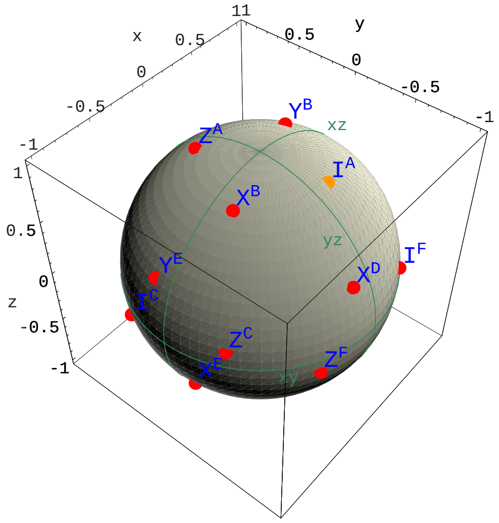


Figure 2.4: The action of the 24 local Clifford operators onto an arbitrarily chosen Bloch vector, namely the orange one. The green circles mark the axes planes. The back side of the sphere is shown in Fig. 2.5

The effect of the Pauli operators X, Y and Z is seen readily: They simply rotate the Bloch vector by 180° about the x, y, and z axis. The S operator, being a square root of Z, is hence a rotation about the z axis by $90°$, where the direction of rotation is according to the right hand rule.[7] This explains the name "phase rotation" for S, as it is the phase which gets transformed into the azimuthal angle ϑ in Eq. (2.12).

What does the Hadamard operator H do? To see that, we build a corresponding matrix acting on the Bloch sphere surface, i. e. an $SO(3)$ matrix, by regarding the action of H onto the three states corresponding to the basis vectors in the $\mathbb{R}^3$. These are mapped as follows:

$$
\begin{array}{ccccccc}
\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \mathrel{\widehat{=}} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \overset{H}{\longmapsto} & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & \mathrel{\widehat{=}} & \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & \mathrel{\widehat{=}} & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix} & \overset{H}{\longmapsto} & \frac{1}{2}\begin{pmatrix} 1+i \\ 1-i \end{pmatrix} & \mathrel{\widehat{=}} & \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} & \mathrel{\widehat{=}} & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & \overset{H}{\longmapsto} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \mathrel{\widehat{=}} & \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}
\end{array}
$$

---

[7] which says that the bent fingers of one's right hand indicate the direction of rotation if one points one's outstretched thumb in the direction of the rotation axis.

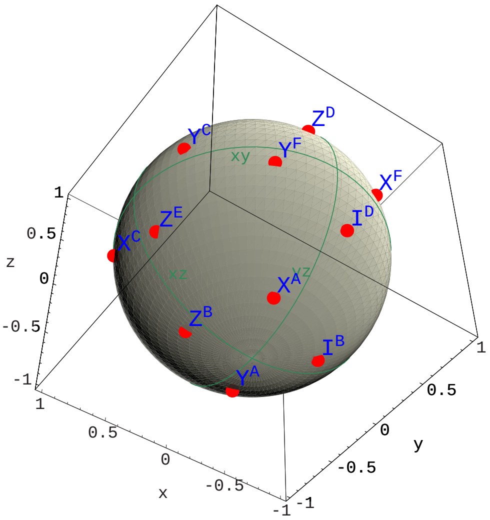


Figure 2.5: The back side of the Bloch sphere of Fig. 2.4

Hence, the two generators of $\mathcal{C}_1$ correspond to the following two $SO(3)$ matrices:

$$\overline{H} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \overline{S} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thus, the Hadamard transform exchanges x and z coordinates on the Bloch sphere. This is in effect a reflection off the half-angle plane between the x and the z axis.

We can now try to visualize the action of all elements of $\mathcal{C}_1$ on the Bloch sphere. A naïve answer would be as follows: We simply concatenate in arbitrary orders the two operations *rotation by 90° degrees about z* ($\overline{S}$) and *reflecting off the x-z half-angle plane* ($\overline{H}$), and see which points on the sphere we can reach after starting from a fixed point for which we choose one that is generic in the sense that is not in fixed sets of the transformation.

Fig. 2.2.5 shows the result. It shows *48* points, twice as many as expected. This is, of course, due to the main difference between $SU(2)$ and $SO(3)$: If one rotates a spin-½ particle once by full 360° around an arbitrary axis, one does not get back to the same state, but picks up a minus sign. The full rotation is not a symmetry operation of the spinor space, but the double rotation (780°) is. The $SO(3)$ group, however, describes rotations in ordinary $\mathbb{R}^3$ space, and hence includes the 360° rotation. Thus, it is to be expected to find the number of symmetry operations doubling, when one goes, via Eq. (2.11), from spinor space $\mathbb{C}^2$ (handled by $SU(2)$) to $\mathbb{R}^3$ (handled by $SO(3)$).

The proper way is to do the described operation in the other order:

Choose an arbitrary state[8] and let all the 24 Clifford operators act on it, *then* transform these 24 states into Bloch vectors with Eq. (2.11) and mark them on the Bloch sphere. The result is shown in Figs. 2.4 and 2.5.

2.2.6 Symplectic formalism

Let us come back to the stabilizer formalism. For certain applications it is useful to represent the n stabilizer generators (without their phases) as n vectors out of the $2n$-bit Boolean field $\mathbb{F}_2^{2n}$. (This formalism has been introduced in [DM03] and is used e. g. in [NDM04b].) In these vectors, a "1" placed at position j and a "0" at position $n+j$ means that the tensor product representation of the Pauli operator contains an X acting on the j-th qubit, i.e. the "1" in the upper half of the vector codes for X operation. Likewise, the lower half codes for Z operations: a "0" at position j and a "1" at $n+j$ means a Z acting on qubit j. Consequently, "0" at both positions means the identity, and "1" at both positions a combined X and Z, i. e. a Y. Note that all phases of the generator matrices are lost when translating to this notation.

To see the effect of a Clifford operation on a stabilizer state, one has to conjugate the stabilizer generators under the Clifford operations. For this purpose, the $\mathbb{F}_2^{2n}$ vectors are left-multiplied with a $2n \times 2n$ matrix over $\mathbb{F}_2$ (satisfying a certain symplecticity condition) which represents the Clifford operation.

For a *local* Clifford operation, acting only on qubit j, this matrix has a simple structure: It is equal to the identity matrix, except at the j-th and the $(n+j)$-th row and column. There, it is all zero, except for the four entries at the intersection of said rows and columns, which form a 2×2 matrix which has to be invertible [NDM04b].

The six 2×2 matrices that are invertible over $\mathbb{F}_2$ correspond to the six permutation symbols:

$$\begin{aligned} Q_A &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & Q_B &= \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \\ Q_C &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, & Q_D &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \\ Q_E &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, & Q_F &= \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}. \end{aligned}$$

The sign symbol cannot be mapped to this formalism as all signs are lost.

---

[8] for the figures, I took, as a generic state,

$$\left(\cos\left(\frac{12\pi}{13}\right), e^{i\frac{3\pi}{5}} \sin\left(\frac{12\pi}{13}\right)\right)^T$$

In the next chapter we shall use the concepts introduced in this chapter for the purpose of simulating quantum circuits.

Chapter 3

PUBLICATION:
Fast simulation of stabilizer circuits using a graph state representation

Simon Anders and Hans J. Briegel

published in *Physical Review A, Vol.* ***73*** *(2006), 022334*

submitted 25 April 2005, published 21 February 2006
DOI: 10.1103/PhysRevA.73.022334

ArXiv preprint: quant-ph/0504117 v2

Abstract

According to the Gottesman-Knill theorem, a class of quantum circuits, namely the so-called stabilizer circuits, can be simulated efficiently on a classical computer. We introduce a new algorithm for this task, which is based on the graph-state formalism. It shows significant improvement in comparison to an existing algorithm, given by Gottesman and Aaronson, in terms of speed and of the number of qubits the simulator can handle. We also present an implementation.

[PACS: 03.67.-a, 03.67.Lx, 02.70.-c]

3.1 Introduction

Protocols in quantum information science often use entangled states of a large number of qubits. A major challenge in the development of such protocols is to actually test them using a classical computer. This is because a straight-forward simulation is typically exponentially slow and hence intractable. Fortunately, the Gottesman-Knill theorem ([Got98a], [NC00]) states that an important subclass of quantum circuits can be simulated efficiently, namely so-called *stabilizer circuits*. These are circuits that use only gates from a restricted subset, the so-called Clifford group. Many techniques in quantum information use only Clifford gates, most importantly the standard algorithms for entanglement purification [BBP$^+$96, DEJ$^+$96, MPP$^+$98, MS02, DAB03] and for quantum error correction [Sho95, Ste96b, CS96, Ste96a]. Hence, if one wishes to study such networks, one can simulate them numerically.

The usual proof of the Gottesman-Knill theorem (as stated e. g. in [NC00]) contains an algorithm that can carry out this task in time $\mathcal{O}(N^3)$, where N is the number of qubits. Especially for the applications just mentioned, one is interested in a large N: For entanglement purification one might want to study large ensembles of states, and for quantum error correction concatenations of codes. The cubic scaling renders this extremely time-consuming, and a more efficient algorithm should be of great use.

Recently, Aaronson and Gottesman presented such an algorithm (and an implementation of it) in Ref. [AG04], whose time and space requirements scale only quadratically with the number of qubits. In the present paper, we further improve on this by presenting an algorithm that for typical applications only requires time and space of $\mathcal{O}(N \log N)$. While Aaronson and Gottesman's simulator, when used on an ordinary desktop computer, can simulate already systems of several thousands of qubits in a reasonable time, we have used our simulator for over a million of qubits. This provides a valuable tool for investigating complex protocols such as our study of multi-party entanglement purification protocols in Ref. [KADB06].

The crucial new ingredient is the use of so-called graph states. Graph states have been introduced in [BR01] for the study of entanglement properties of certain multi-qubit systems; they were used as starting point for the one-way quantum computer (i. e., measurement-base quantum computing) [RBB03], and found to be suited to give a graphical description of CSS codes (for quantum error correction) [SW02]. Graph states take their name from the concept of graphs in mathematics: Each qubit corresponds to a vertex of the graph, and the graph's edges indicate which qubits have interacted (see below for details).

There is an intimate correspondence between stabilizer states (the class of states that can appear in a stabilizer circuit) and graph states: Not only is every graph state a stabilizer state, but also every stabilizer state is equivalent to a graph state in the following sense: Any stabilizer state can be transformed to a graph state by applying a tensor product of local Clifford (LC) operations [Sch01, GKR02, NDM04b]. We shall call these local Clifford operators the *vertex operators* (VOPs).

To represent a stabilizer state in computer memory, one stores its tableau of stabilizer operators, which is an $N \times N$ matrix of Pauli operators and hence takes space of order $\mathcal{O}(N^2)$ (see below for details). Gottesman and Aaronson's simulator extends this matrix by another matrix of the same size (which they call the destabilizer tableau), so that their simulator has space complexity $\mathcal{O}(N^2)$. A graph state, on the other hand, is described by a mathematical graph, which, for reasons argued later, only needs space of $\mathcal{O}(N \log N)$ in typical applications. Hence, much larger systems can be represented in memory, if one describes them as graph states, supplemented with the list of VOPs. However, we also need efficient ways to calculate how this representation changes, when the represented state is measured or undergoes a Clifford gate application. The effect of measurements has been extensively studied in [HEB04], and gate application is what we will study in this paper, so that we can then assemble both to a simulation algorithm.

This paper is organized as follows: We first review the stabilizer formalism, the Gottesman-Knill theorem, and the graph state formalism in Section 3.2. There, we will also explain our representation in detail. Section 3.3 explains how the state representation changes when Clifford gates are applied. This is the main result and the most technical part of the paper. For the simulation of measurements, we can rely on the studies of Ref. [HEB04], which are reviewed and applied for our purpose in Section 3.4. Having exposed all parts of the simulator algorithm, we continue by presenting our implementation of it. A reader who only wishes to use our simulator and is not interested in its internals may want to read only this section. Section 3.6 assesses the time requirements of the algorithm's components described in Sections 3.3 and 3.4 in order to prove our claim of superior scaling of performance. We finish with a conclusion (Section 3.7).

3.2 Stabilizer and graph states

We start by explaining the concepts mentioned in the introduction in a formal manner.

Definition 3.1 *The Clifford group $\mathcal{C}_N$ on N qubits is defined as the normalizer of the Pauli group $\mathcal{P}_N$:*

$$\mathcal{C}_N = \left\{ U \in SU(2^N) \mid UPU^\dagger \in \mathcal{P}_N \quad \forall P \in \mathcal{P}_N \right\}, \qquad \mathcal{P}_N = \{\pm 1, \pm i\} \cdot \{I, X, Y, Z\}^{\otimes N}, \quad (3.1)$$

where I is the identity and X, Y, and Z are the usual Pauli matrices.

The Clifford group can be generated by three elementary gates (see e. g. [NC00]): the Hadamard gate H, the $\frac{\pi}{4}$ phase rotation S, and a two-qubit gate, either the controlled NOT gate ΛX, or the controlled phase gate ΛZ:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$\Lambda X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \Lambda Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{3.2}$$

The significance of the Clifford group is due to the Gottesman-Knill theorem ([Got98a], see also [NC00]):

Theorem 3.1 *A quantum circuit using only the following elements (called a stabilizer circuit) can be simulated efficiently on a classical computer:*

- *preparation of qubits in computational basis states*
- *quantum gates from the Clifford group*
- *measurements in the computational basis*

The proof of the theorem is simple after one introduces the notion of stabilizer states [Got97]:

Definition 3.2 *An N-qubit state $|\psi\rangle$ is called a stabilizer state if it is the unique eigenstate with eigenvalue +1 of N commuting multi-local Pauli operators P_a (called the stabilizer generators):*

$$P_a\,|\psi\rangle = |\psi\rangle\,, \quad P_a \in \mathcal{P}_N, \quad a = 1,\dots,N$$

(These N operators generate an Abelian group, the *stabilizer*, of 2^N Pauli operators that all satisfy this stabilization equation.)

Computational basis states are stabilizer states. Furthermore, if a *Clifford* gate U acts on a stabilizer state $|\psi\rangle$, the new state $U\,|\psi\rangle$ is a stabilizer state with generators $UP_iU^\dagger \in \mathcal{P}_N$. Hence, the state in a stabilizer circuit can always be described by the *stabilizer tableau*, which is a matrix of $N \times N$ operators from $\{I, X, Y, Z\}$ (where each row is preceded by a sign factor). The effect of an n-qubit gate can then be determined by updating nN elements of the matrix, which is an efficient procedure.

Instead of on the stabilizer tableau, we shall base our state representation on graph states:

Definition 3.3 *An N-qubit graph state $|G\rangle$ is a quantum state associated with a mathematical graph $G = (V, E)$, whose $|V| = N$ vertices correspond to the N qubits, while the edges E describe quantum correlations, in the sense that $|G\rangle$ is the unique state satisfying the N eigenvalue equations*

$$K_G^{(a)}\,|G\rangle = |G\rangle\,, \quad a \in V,$$
$$\text{with } K_G^{(a)} = \sigma_x^{(a)} \prod_{b\in\mathrm{ngbh}\,a} \sigma_z^{(b)} =: X_a \prod_{b\in\mathrm{ngbh}\,a} Z_b, \tag{3.3}$$

where ngbh $a := \{b \mid \{a,b\} \in E\}$ *is the set of vertices adjacent to* a *[RBB03, BR01, SW02].*

The following theorem states that the edges of the graph can be associated with phase gate interactions between the corresponding qubits:

Theorem 3.2 *If one starts with the state* $|+\rangle^{\otimes N} = \prod_{a \in V} H_a |00\ldots0\rangle$ *one can easily construct* $|G\rangle$ *by applying* ΛZ *on all pairs of neighboring qubits:*

$$|G\rangle = \left(\prod_{\{a,b\} \in E} \Lambda Z_{ab} \right) \left(\prod_{a \in V} H_a \right) |0\rangle^{\otimes N} \tag{3.4}$$

(Proof: Insert Eq. (3.4) into Eq. (3.3) [HEB04].)

As the operators $K_G^{(a)}$ belong to the Pauli group, all graph states are stabilizer states, and so are the states which we get by applying *local* Clifford operators $C \in \mathcal{C}_1$ to $|G\rangle$. For such states, we introduce the notation

$$|G; \underline{C}\rangle := |G; C_1, C_2, \ldots, C_N\rangle := \bigotimes_{i=1}^{N} C_i |G\rangle \tag{3.5}$$

It has been shown that all stabilizer states can be brought into this form [Sch01, GKR02, NDM04b], i. e. any stabilizer state is LC-equivalent to a graph state. (We call two states LC-equivalent if one can be transformed into the other by applying a tensor product of local Clifford operators.) Finding the graph state that is LC-equivalent to a stabilizer state given by a tableau can be done by a sort of Gaussian elimination as explained in [NDM04b].

This is what we shall use to represent the current quantum state in the memory of our simulator. Fig. 3.1 shows for an example state the tableau representation that is usually employed (and also used by CHP, albeit in a modified form) and our representation. The tableau representation requires space of order $\mathcal{O}(N^2)$. We store the graph in adjacency list form (i. e., for each vertex, a list of its neighbors is stored), which needs space of order $\mathcal{O}(N\bar{d})$, where $\bar{d}$ is the average vertex degree (number of neighbors) in the graph. We also store a list of the N local Clifford operators $C_1, \ldots, C_N$, which transform the graph state $|G\rangle$ into the stabilizer state $|G; \underline{C}\rangle$. We call these operators the vertex operators (VOPs). As there are only 24 elements in the local Clifford group, each VOP is represented as a number in $0, \ldots, 23$. The scheme to enumerate the 24 operators will be described in [·][1]. Note that we can disregard global phases of the VOPs as they only lead to a global phase of the full state of the simulator.

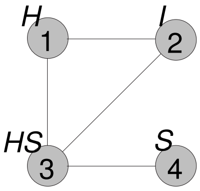
---

[1] In the journal publication, a reference to "S. Anders, in preparation" was given here. I regret that this information has not been published separately so far, but can now be found as Sec. 2.2 of the present thesis.

(a)

	1	2	3	4
$+$	Z	Z	X	I
$+$	X	X	X	I
$-$	X	Z	Y	Z
$+$	I	I	X	Y

(b)



(c)

Vertex	VOP	adjacency list
1	10	2, 3
2	0	1, 3
3	17	1, 2, 4
4	6	3

Figure 3.1: A stabilizer state $|\psi\rangle$ represented in different ways: (a) as stabilizer tableau, i. e. the state is stabilized by the group of Pauli operators generated by the operators in the 4 rows. This representation needs space $\mathcal{O}(N^2)$ for N qubits. (b), (c) as LC-equivalence to a graph state. (b) shows the graph, with the VOPs given by their decomposition into the group generators $\{H, S\}$. (c) is the data structure that represents (b) in our algorithm. The VOPs are now specified using numbers between 0 and 23 (which enumerate the $|\mathcal{C}_1| = 24$ LC operators). Here, we need space $\mathcal{O}(N\overline{d})$, where $\overline{d}$ is the average vertex degree, i. e. the average length of the adjacency lists. Writing G for the graph in (b), we can use the notation of Eq. (3.5) and write $|\psi\rangle = |G; H, I, HS, S\rangle$.

As we shall see later, we may typically assume that $\overline{d} = \mathcal{O}(\log N)$. Hence, our representation needs considerably less space in memory than a tableau, namely $\mathcal{O}(N \log N)$, including $\mathcal{O}(N)$ for the VOP list.

The Gaussian elimination needed to transform a stabilizer tableau to its graph state representation is slow (time complexity $\mathcal{O}(N^3)$), and so we should better not use it in our simulator. But usually, one starts with the initial state $|0\rangle^{\otimes N}$, and if we write this state already in graph state form, the tableau representation is never used at all.

From Eq. (3.4), it is clear that the initial state can be written as a graph with no edges and Hadamard gates acting on all vertices:

$$|0\rangle^{\otimes N} = |(\{1, \dots, N\}, \{\}); H, \dots, H\rangle\,.$$

3.3 Gates

When the simulator is asked to simulate a Clifford gate, the current stabilizer state is changed and its graph representation has to be updated to correctly reflect the action of the gate. How to do this, is the main technical result of this paper.

Single-qubit gates

In the graph representation, applying local (single-qubit) Clifford gates becomes trivial: if $C \in \mathcal{C}_1$ is applied to qubit a, we replace this qubit's VOP C_a by CC_a.

Two-qubit gates

It is sufficient if the simulator is capable to simulate a single multi-qubit gate: As the entire Clifford group is generated, e. g., by H, S, and ΛZ, all gates can be constructed by concatenating these. We chose to implement ΛZ, the phase gate, as this is (because of its role in Eq. (3.4)) most natural for the graph-state formalism.

In the following discussion, the two qubits onto which the phase gate acts, are called the *operand vertices* and denoted with a and b. All other qubits are called *non-operand vertices* and denoted $c, d, \ldots$.

To solve the task, we have to distinguish several cases.

Case 1. *The VOPs of both operand vertices are in $\mathcal{Z}$*, where $\mathcal{Z} := \{I, Z, S, S^\dagger\}$ denotes the set of those four local Clifford operators that commute with ΛZ (the other 20 operators do not). In this case, applying the phase gate is simple: We use the fact that (due to Eq. (3.4)) applying a phase gate on a graph state just toggles an edge:

$$\Lambda Z_{ab} \left|(V, E)\right\rangle = \left|(V, E \vartriangle \{\{a, b\}\})\right\rangle ,$$

where $\vartriangle$ denotes the symmetric set difference $A \vartriangle B := (A \cup B) \setminus (A \cap B)$, i. e. the edge $\{a, b\}$ is added to the graph if is was not present before, otherwise it is removed.

Case 2. *The VOP of at least one of the operand vertices is not in $\mathcal{Z}$.* In this case, just toggling the edge is not allowed because the ΛZ_{ab} cannot be moved past the non-$\mathcal{Z}$ VOP. But there is a way to change the VOPs without changing the state, which works in the following case:

Sub-case 2.2. *Both operand vertices have non-operand neighbors*. Here, the following operation will help:

Definition 3.4 *The operation of local complementation about a vertex a of a graph $G = (V, E)$, denoted L_a, is the operation that inverts the subgraph induced by the neighborhood of v:*

$$L_a(V, E) = (V, E \vartriangle \{\{b, c\} | b, c \in \operatorname{ngbh} a\})$$

This operation transforms the state into a local-Clifford equivalent one, as the following theorem, taken from [HEB04, NDM04b], asserts:

Theorem 3.3 *Applying the local complementation L_a onto a graph G yields a state $|L_a G\rangle = U\,|G\rangle$, with the multi-local unitary*

$$U = \sqrt{-iX_a} \prod_{b\in \operatorname{ngbh} a} \sqrt{iZ_b} \propto \sqrt{K_G^{(a)}}.$$

Note that the operator $\sqrt{iZ}$ is related to the phase operator S of Eq. (3.2): $\sqrt{iZ} = e^{i\frac{\pi}{4}} S^\dagger$, and $\sqrt{iX} = \sqrt{-iX}^\dagger = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$.

An obvious consequence of Theorem 3.3 is the following.

Corollary 3.1 *A state $|G; \underline{C}\rangle$ is invariant under application of L_a to G, followed by an updating of C according to*

$$C_b \mapsto \begin{cases} C_b\sqrt{iX} & \text{for } b = a \\ C_b\sqrt{-iZ} & \text{for } b \in \operatorname{ngbh} a \\ C_b & \text{otherwise} \end{cases}. \tag{3.6}$$

Now note that the local Clifford group is generated not only by S and H but also by $\sqrt{-iX}$ and $\sqrt{iZ}$, the Hermitean adjoints of the operators right-multiplied to the VOPs in Eq. (3.6). Our simulator has a look-up table that spells out every local Clifford operator as a product of –as it turns out, at most 5– of these two operators, times a disregarded global phase. For example, the table's line for H reads:

$$H \propto \sqrt{-iX}\sqrt{iZ}\sqrt{iZ}\sqrt{iZ}\sqrt{-iX}. \tag{3.7}$$

This allows us now to reduce the VOP C_a of any *non-isolated* vertex a to the identity I by proceeding as follows: The decomposition of C_a taken from the look-up table is read from right to left. When a factor $\sqrt{-iX}$ is read we do a local complementation about a. This does not change the state if the correction of Eq. (3.6) is applied, which right-multiplies a factor $\sqrt{iX}$ to C_a. This factor $\sqrt{iX}$ cancels with the factor $\sqrt{-iX}$ at the right-hand end of C_a's decomposition, so that we now have a VOP with a shorter decomposition.

If the right-most operator of the decomposition is $\sqrt{iZ}$ we do a local complementation about an arbitrarily chosen neighbor of a, called a's "swapping partner". Now, the correction operation will lead to a factor S being right-multiplied to C_a, again shortening the decomposition.

Note that a local complementation about a never changes the edges incident on a and hence, if a was non-isolated in the beginning of the procedure, it will stay so. This is important, as only a non-isolated vertex can have a swapping partner. Hence, the procedure can be iterated, and (as the decompositions have a maximum length of 5) after at most 5 iterations, we are left with the identity I as VOP.

We apply the described “VOP reduction procedure” to both operand vertices. After that, both vertices are the identity, and we can proceed as in Case 1.

One might wonder, however, whether the use of the VOP reduction procedure on the second operand vertex b spoils the reduction of the VOP of the first operand a. After all, a could be a neighbor of b or of the swapping partner c of b. Then, if a local complementation L_b or L_c is performed, the compensation according to Eq. (3.6) changes the neighborhood of b and c (which include a). But note that a neighbor of the inversion center only gets a factor $\sqrt{-iZ} \propto S^\dagger$. As $S^\dagger$ generates $\mathcal{Z}$, this means that after the reduction of b, the VOP of a might be no longer the identity but it is still an element of $\mathcal{Z}$, and we are allowed to go on with Case 1.

But what happens, if one of the vertices does not have a non-operand neighbor, that could serve as swapping partner? This is the next Sub-case.

Sub-case 2.2. *At least one of the operand vertices is isolated or only connected to the other operand vertex.* We first assume that the other vertex is non-connected in the same sense:

Sub-sub-case 2.2.1. *Both operand vertices are either completely isolated, or only connected with each other.* Then, we can ignore all other vertices and have to study only a finite, rather small number of possible states.

Let us denote by $\bullet\ \bullet$ the 2-vertex graph with no edges, and by $\bullet\!\!-\!\!\bullet$ the 2-vertex graph with one edge. There are only very few possible 2-qubit stabilizer states, namely those in

$$\mathcal{S}_2 := \{|G; C_1, C_2\rangle \mid G \in \{\bullet\ \bullet, \bullet\!\!-\!\!\bullet\}, C_1, C_2 \in \mathcal{C}_1\}\,. \tag{3.8}$$

Of course, many of the assignments in the r.h.s describe the same state, such that $|\mathcal{S}_2| < 2 \cdot 24^2$. Remember that the phase gate $\Lambda Z_{1,2}$ (being a Clifford operator) maps $\mathcal{S}_2$ bijectively onto itself.

The function table of $\Lambda Z_{1,2}|_{\mathcal{S}_2} : |G; C_1, C_2\rangle \mapsto |G'; C_1', C_2'\rangle$ can easily be computed in advance (we did it with Mathematica) and hard-coded into the simulator as a look-up table. This table contains $2 \cdot 24^2$ lines such as

$$|\bullet\ \bullet, C_{[13]}, C_{[2]}\rangle \mapsto |\bullet\!\!-\!\!\bullet, C_{[0]}, C_{[2]}\rangle\,, \tag{3.9}$$

where the $C_{[i]}(i = 0, \ldots, 23)$ are the Clifford operators in the enumeration detailed in [Sec. 2.2] (e. g. $C_{[0]} = I$, $C_{[2]} = Y$).

Note that many of the assignments to C_1 and C_2 in Eq. (3.8) describe the same state. Hence, we have a choice in the operators C_1', C_2' with which we represent the results of the phase gate in the look-up table. It turns out (by inspection of all the possibilities) that we can always choose the operators such that the following constraint is fulfilled:

Constraint 1. *If* $C_1(C_2) \in \mathcal{Z}$, *choose* C_1', C_2' *such that again* $C_1'(C_2') \in \mathcal{Z}$. The use of this will become clear soon.

Sub-case 2.2.2. We are left with one last case, namely that one vertex, let it be a, is connected with non-operand neighbors, but the other vertex b is not, i. e. has either no neighbors or only a as neighbor. Then, we proceed as follows: We use iterated local complementations to reduce C_a to I. After that, we may use the look-up table as in Sub-sub-case 2.2.1. That this is allowed even though a is connected to a non-operand vertex is shown in the following: First note that the state after the reduction of C_a to I can be written (following Eq. (3.5)) as

$$|(V,E);\underline{C}\rangle = \prod_{c\in V} C_c \prod_{\{c,d\}\in E} \Lambda Z_{cd}\ |{+}{+}\cdots{+}\rangle$$

$$= \underbrace{\prod_{\substack{c\in\\ V\setminus\{a,b\}}} C_c \prod_{\substack{\{c,d\}\in\\ E\setminus\{\{a,b\}\}}} \Lambda Z_{cd}}_{\substack{C_b \text{ and } \Lambda Z_{ab}\\ \text{commute with this}}}\ \underbrace{C_b\,(\Lambda Z_{ab})^{\zeta}\,|{+}{+}\cdots{+}\rangle}_{\substack{=|+\rangle^{\otimes N-2}\otimes|\varphi\rangle_{ab}\\ \text{with } |\varphi\rangle\in\mathcal{S}_2\\ (\star)}}$$

(where $\zeta = 0, 1$ indicates whether $\{a, b\} \in E$). Observe that C_b has been moved past the operators ΛZ_{cd}. This is allowed because none of the ΛZ_{cd} acts on b

We now apply ΛZ_{ab} to this state. ΛZ_{ab} can be moved through all the phase gates and vertex operators above the left brace so that it stands right in front of the $\mathcal{S}_2$ state $|\varphi\rangle_{ab}$ which is separated from the rest. Thus, the table (3.9) from Sub-sub-case 2.2.1 may be used. (This would not be the case if, in the state above the brace marked with "$(\star)$", the two operand vertices were still entangled with other qubits.) The table look-up will give new operators C_a', C_b' and a new ζ', so that the new state has the following form:

$$\Lambda Z_{ab}\,|(V,E);\underline{C}\rangle = \prod_{\substack{c\in\\ V\setminus\{a,b\}}} C_c \prod_{\substack{\{c,d\}\in\\ E\setminus\{\{a,b\}\}}} \Lambda Z_{cd}\ C_a' C_b'\,(\Lambda Z_{ab})^{\zeta'}\,|{+}{+}\cdots{+}\rangle \qquad (3.10)$$

For this to be a state in our usual $|G;\underline{C}\rangle$ form (3.5), the two operators C_a' and C_b' have to moved to the left, through the ΛZ_{cd}. For C_b', this is no problem, as b was assumed to be either isolated or connected only to a, so that C_b' commutes with $\prod_{\{c,d\}\in E\setminus\{\{a,b\}\}} \Lambda Z_{cd}$, as the latter operator does not act on b. The vertex a, however, has connections to non-operand neighbors, so that some of the ΛZ_{cd} act on it. We may move it only if $C_a' \in \mathcal{Z}$ (as this means that it commutes with ΛZ). Luckily, due to Constraint 1 imposed above, we can be sure that $C_a' \in \mathcal{Z}$, because $C_a = I \in \mathcal{Z}$.

```
1  cphase (vertex $a$, vertex $b$):
2      if ngbh $a \setminus \{b\} \neq \{\}$:
3          remove_VOP $(a, b)$
4      end if
5      if ngbh $b \setminus \{a\} \neq \{\}$:
6          remove_VOP $(b, a)$
7      end if
8      [It may happen that the condition in line 2 has not been fulfilled then,
       but is now due to the effect of line 5. So we check again:]
9      if ngbh $a \setminus \{b\} \neq \{\}$:
10         remove_VOP $(a, b)$
11     end if
12     [Now we can be sure that the the condition $(\text{ngbh}\, c \setminus \{a, b\} = \{\}$ or
       $\mathit{VOP}[c] \in \mathcal{Z})$ is fulfilled for $c = a, b$ and we may use the lookup table (cf.
       Eq. (3.9)).]
13     if $\{a, b\} \in E$ :
14         edge $\longleftarrow$ true
15     else:
16         edge $\longleftarrow$ false
17     end if
18     $(\texttt{edge}, \texttt{VOP}[a], \texttt{VOP}[b]) \longleftarrow$ cphase_table$[\texttt{edge}, \texttt{VOP}[a], \texttt{VOP}[b]]$
```

Listing 3.1a: Pseudo-code for controlled phase gate (ΛZ) acting on vertices a and b (`cphase`) [here], and for the two auxiliary routines `remove_VOP` [Listing 3.1b] and `local_complementation` [Listing 3.1c].

Listing 3.1 shows in pseudo-code how these results can be used to actually implement the controlled phase gate ΛZ.

3.4 Measurements

In a stabilizer circuit, the simulator may be asked at any point to simulate the measurement of a qubit in the computational basis. How the outcome of the measurement is determined, and how the graph representation has to be updated in order to then represent the post-measurement state will be explained in the following.

To measure a qubit a of a state $|G, \underline{C}\rangle$ in the computational basis means to measure the qubit in the underlying graph state $|G\rangle$ in one of the 3 Pauli

```
remove_VOP (vertex $a$, vertex $b$):
[This reduces VOP[$a$] to $I$, avoiding (if possible) to use $b$ as swapping partner.]
    [First, we choose a swapping partner $c$.]
    if ngbh $a \setminus \{b\} \neq \{\}$:
        $c \leftarrow$ any element of ngbh $a \setminus \{b\}$
    else:
        $c \leftarrow b$
    end if
    $d \leftarrow$ decomposition_lookup_table $[a]$
    [$c$ contains now a decomposition such as Eq. (3.7)]
    for $v$ from last factor of $d$ to first factor of $d$
        if $v = \sqrt{-iX}$:
            local_complementation ($a$)
        else: (this means that $v = \sqrt{iZ}$)
            local_complementation ($b$)
        end if
    [Now, VOP[$a$] $= I$.]
```

Listing 3.1b

```
local_complementation (vertex $a$)
[performs the operation specified in Definition 3.4]
    $n_v \leftarrow$ ngbh $v$
    for $i \in n_v$:
        for $j \in n_v$:
            if $i < j$:
                if $(i, j) \in E$:
                    remove edge $(i, j)$
                else:
                    add edge $(i, j)$
                end if
            end if
        end for
        VOP$[i] \leftarrow$ VOP$[i]\sqrt{-iZ}$
    VOP$[v] \leftarrow$ VOP$[v]\sqrt{iX}$
    end for
```

Listing 3.1c

bases. Writing the measurement outcome as ζ, this means:

$$\frac{I+(-1)^{\zeta}Z_a}{2}\left|G,\underline{C}\right\rangle = \left(\prod_{b\in V\setminus\{a\}} C_b\right)\frac{I+(-1)^{\zeta}Z_a}{2}C_a\left|G\right\rangle$$
$$= \left(\prod_{b\in V\setminus\{a\}} C_b\right) C_a\frac{I+(-1)^{\zeta}C_a^{\dagger}Z_aC_a}{2}\left|G\right\rangle \quad (3.11)$$

As C_a is a Clifford operator, $P_a := C_a^{\dagger}Z_aC_a \in \{X_a, Y_a, Z_a, -X_a, -Y_a, -Z_a\}$. Thus, in order to measure qubit a of $\left|G,\underline{C}\right\rangle$ in the *computational* basis, we measure the observable P_a on $\left|G\right\rangle$. Note that in case that P_a is the negative of a Pauli operator, the measurement result ζ to be reported by the simulator is the complement of $\tilde{\zeta}$, the result given by the X, Y or Z measurement on the underlying graph state $\left|G\right\rangle$.

How is the graph G changed and how do the vertex operators have to be modified if the measurement $\frac{I\pm P_a}{2}\left|G\right\rangle$ is carried out? This has been worked out in detail in Ref. [HEB04], which we now briefly review for the present purpose.

The simplest case is that of $P = \pm Z$. Here, the state changes as follows:

$$\frac{I+(-1)^{\tilde{\zeta}}Z_a}{2}\left|(V,E)\right\rangle =$$
$$\underbrace{\left(X_a\prod_{b\in\mathrm{ngbh}\,a} Z_b\right)^{\tilde{\zeta}}}_{(\star)} H_a\left|(V,E\setminus\{\{a,b\}|b\in\mathrm{ngbh}\,a\})\right\rangle. \quad (3.12)$$

The value of $\tilde{\zeta}$ is chosen at random (using a pseudo-random number generator). To update the simulator state, the VOPs are right-multiplied with the under-braced operators $(\star)$ and the edges incident on a are deleted as indicated in the ket.

A measurement of the Y observable ($P = \pm Y$) requires a complementation of the edges set according to

$$E \mapsto E \bigtriangleup \{\{b,c\} \mid b,c\in\mathrm{ngbh}\,a\}$$

and a change in the VOPs as follows:

$$C_b \mapsto C_b\sqrt{-iZ}^{(\dagger)} \text{ for } b\in\mathrm{ngbh}\,a\cup\{a\},$$

where the dagger in parentheses is to be read only for measurement result $\tilde{\zeta} = 1$.

The most complicated case is the X measurement which requires an update of edges and VOPs as follows:

$$\begin{aligned} E \mapsto E &\bigtriangleup \{\{c,d\} \mid c \in \operatorname{ngbh} b,\, d \in \operatorname{ngbh} a\} \\ &\bigtriangleup \{\{c,d\} \mid c,d \in \operatorname{ngbh} b \cap \operatorname{ngbh} a\} \\ &\bigtriangleup \{\{b,d\} \mid d \in \operatorname{ngbh} a \backslash \{b\}\} \end{aligned}$$

$$C_c \mapsto \begin{cases} C_c Z^{\tilde{\zeta}} & \text{for } c = a \\ C_c \sqrt{iY}^{(\dagger)} & \text{for } c = b \quad (\text{read “}{\dagger}\text{” only for } \tilde{\zeta} = 1) \\ C_c Z & \text{for } c \in \begin{cases} \operatorname{ngbh} a \setminus \operatorname{ngbh} b \setminus \{b\} \\ \qquad (\text{for } \tilde{\zeta} = 0) \\ \operatorname{ngbh} b \setminus \operatorname{ngbh} a \setminus \{a\} \\ \qquad (\text{for } \tilde{\zeta} = 1) \end{cases} \\ C_c & \text{otherwise} \end{cases} \tag{3.13}$$

Here, b is a vertex chosen arbitrarily from ngbh a and $\sqrt{iY} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$.

In all these cases the measurement result is chosen at random. Only in case of the measurement of $P_a = \pm X$ an isolated vertex, the result is always $\tilde{\zeta} = 0$ (which means an actual result of $\zeta = 0$ for $P_a = X$ and $\zeta = 1$ for $P_a = -X$.)

3.5 Implementation

The algorithm described above has been implemented in C++ in object-oriented programming style. We have used the GNU Compiler Collection (GCC) [GCC] under Linux, but it should be easy to compile the program on other platforms as well.[2] The implementation is done as a library to allow for easy integration into other projects. We also offer bindings to Python [R+], so that the library can be used by Python programs as well. (This was achieved using SWIG [B+].)

The simulator, called “GraphSim” can be downloaded from `http://homepage.uibk.ac.at/homepage/c705/c705213/work/graphsim.html`.

A detailed documentation of the library is supplied with it. To demonstrate the usage here at least briefly, we give Listing 2 as a simple toy example. It is written in Python, and a complete program.

In the example, we start by loading the GraphSim library (Line 2) and then initialize a register of 8 qubits (line 4), which are then all in $|0\rangle$ state. We get an object called “`gr`” of class `GraphRegister`, which represents the

---

[2] We use only ISO Standard C++ with one exception: The `hash_set` template is used, which is, though not part of the standard, supplied by most modern compilers.

```
import random
import graphsim

gr = graphsim.GraphRegister (8)

gr.hadamard (4)
gr.hadamard (5)
gr.hadamard (6)
gr.cnot (6, 3)
gr.cnot (6, 1)
gr.cnot (6, 0)
gr.cnot (5, 3)
gr.cnot (5, 2)
gr.cnot (5, 0)
gr.cnot (4, 3)
gr.cnot (4, 2)
gr.cnot (4, 1)

for i in xrange (7):
   gr.cnot (i, 7)

print gr.measure (7)

gr.print_stabilizer ()
```

Listing 3.2: A simple example in Python

register of qubits. For all following operations, we use the methods of `gr` to access its functionality. In our example, we simply build up an encoded "0" state in the well-known 7-qubit Steane code, which we then measure.

First, we apply Hadamard and CNOT gates onto the qubits with number 0 through 6 in order to build up the Steane-encoded "0" (Lines 6–17). To check that we did so, we measure the encoded qubit, which is done by using CNOT gates to sum up their parity in the eighth qubit ("qubit 7") (Lines 19, 20). Measuring qubit 7 then gives "0", as it should (Line 22).

For further details on using of the GraphSim library from a C++ or Python program, please see the documentation supplied with the source code.

With approximately 1400 lines, GraphSim is complex enough that one cannot take for granted that it faithfully implements the described algorithm without bugs, and testing is necessary. Fortunately, this can be done very conveniently by comparing with Aaronson and Gottesman's "CHP" simulator. As these two programs use quite different algorithms to do the same task, it is very unlikely that any bugs, which they might have, produce the *same* false results. Hence, if both programs give the same result, they can reasonably be considered both to be correct.

We set up a script to do random gates and measurements on a set of qubits for millions of iterations. All operations were performed simultaneously with CHP and GraphSim. For measurements whose outcome was chosen at random by CHP, a facility of GraphSim was used that overrides the random choice of measurement outcomes and instead uses a supplied value. For measurements with determined outcome, however, it was checked whether both programs output the same result. Also, every 1000 steps, the stabilizer tableau of GraphSim's state was calculated from its graph representation and compared to CHP's tableau.[3]

After simulation $4 \cdot 10^6$ operations on 200 qubits in 18 hours and $2 \cdot 10^8$ operations on 20 qubits in 19.7 hours without seeing discrepancies, we are confident that we have exhausted all special cases, so that the two programs can be assumed to always give the same output. As they are based on very different algorithm, this reasonably allows to conclude that they both operate correctly.

3.6 Performance

We now show that our simulator yields the promised performance, i. e. performs a simulation of M steps in time of order $\mathcal{O}(NdM)$, where N is the

---

[3] This was done with a Mathematica subroutine which tries to find a row adding and swapping arrangement to transform one tableau into the other.

number of qubits and d the maximum vertex degree that is encountered during the calculation. Let us go through the different possible simulation steps in order to assess their respective time requirements.

Single-qubit gates are fastest: they only need one look-up in the multiplication table of the local Clifford group (which is hard-coded into the simulator), and are hence of time complexity $\Theta(1)$.

Measurements have a complexity depending on the basis in which they have to be carried out. For a Z measurement, we have to remove the $\deg a$ edges of the measured vertex a. As d is the maximum vertex degree that is to be expected within the studied problem, the complexity of a Z measurement is $\mathcal{O}(d) \leq \mathcal{O}(N)$ (as $d \leq N$).

For a Y and X measurement, we have to do local complementation, which requires dealing with up to $\frac{d(d-1)}{2}$ edges, and hence, the overall complexity of measurements is $\mathcal{O}(d^2)$.

For the phase gate, the same holds. Here, we need a fixed number (up to 5) of local complementations. Thus, measurements and two-qubit gates take $\mathcal{O}(d^2)$ time.

This would be no improvement to Aaronson and Gottesman's algorithm, if we had $d = \mathcal{O}(N)$. The latter is indeed the case if one applies randomly chosen operations as we did to demonstrate GraphSim's correctness. There, we indeed did not observe any superiority in run-time of GraphSim.

In practice, however, this is quite different. For example, when simulating quantum error correction, one can reasonable assume $d = \mathcal{O}(\log N)$. This is because all QEC schemes avoid to do to many operations on one and the same qubit in a row, as this would spread errors. So, vertex degrees remain small. The same reasoning applies to entanglement purification schemes and, more generally, to all circuits which are designed to be robust against noise.

The space complexity is dominated by the space needed to store the quantum state representation. As argued in Section 3.2, this requires only space of $\mathcal{O}(N\bar{d})$, where $\bar{d}$ is the average vertex degree. As explained above, we may expect $\bar{d}$ (as d) to scale sub-linearly with N in typical application, in many applications as $\mathcal{O}(N \log N)$. This is what allows us to handle substantially more qubits than it is possible with the $\mathcal{O}(N^2)$ tableau representation.

As a first practical test, we used GraphSim to simulate entanglement purification of cluster states with the protocol of Ref. [DAB03]. This has been a starting point of a detailed analysis of the communication costs of establishing multipartite entanglement states via noisy channels [KADB06]. Fig. 3.2 demonstrates that GraphSim is indeed suitable for this purpose. Note, that for the right-most data points, the register holds 30,000 qubits.

As we did a Monte Carlo simulation, we had to loop the calculation very often and still got an output within a few hours. For simulations involving several millions of qubits and a large number of runs, we waited about a week for the results when using eight processors in parallel. We redid some

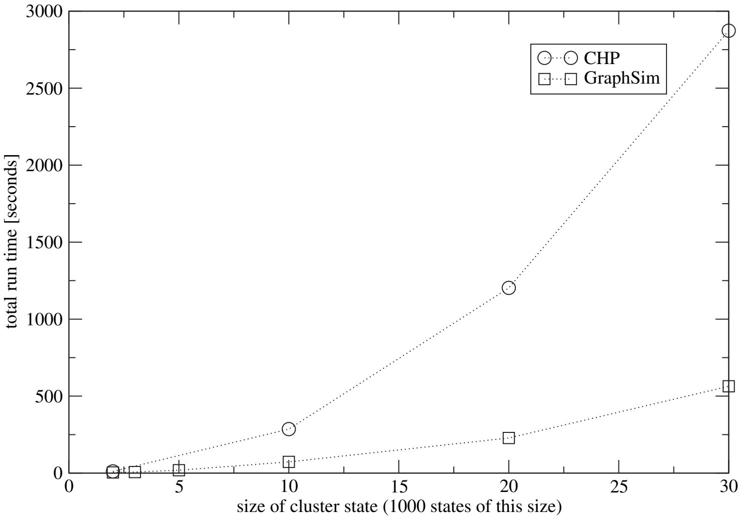


Figure 3.2: Comparison of the performance of CHP and GraphSim. A simulation of entanglement purification was used as sample application. The register has 1000 times the size of the states to hold an ensemble of 1000 states.

of these calculations in a more controlled testing environment as a benchmark for GraphSim. Fig. 3.3 shows the results in a log-log plot.

3.7 Conclusion

To summarize, we have used recent results on graph states to find a very space-efficient representation of stabilizer states, and determined, how this representation changes under the action of Clifford gates. This can be used to simulate stabilizer circuits more efficiently than previously possible. The gain is not only in simulation speed, but also in the number of manageable qubits. In the latter, at least two orders of magnitude are gained. We have presented an implementation of our simulation algorithm and will soon publish results about entanglement purification which makes use of our new technique.

Acknowledgments

We would like to thank Marc Hein for most helpful discussions.

This work was supported in part by the Austrian Science Foundation (FWF), the Deutsche Forschungsgemeinschaft (DFG), and the European Union (IST-2001-38877, -39227, OLAQUI, SCALA).

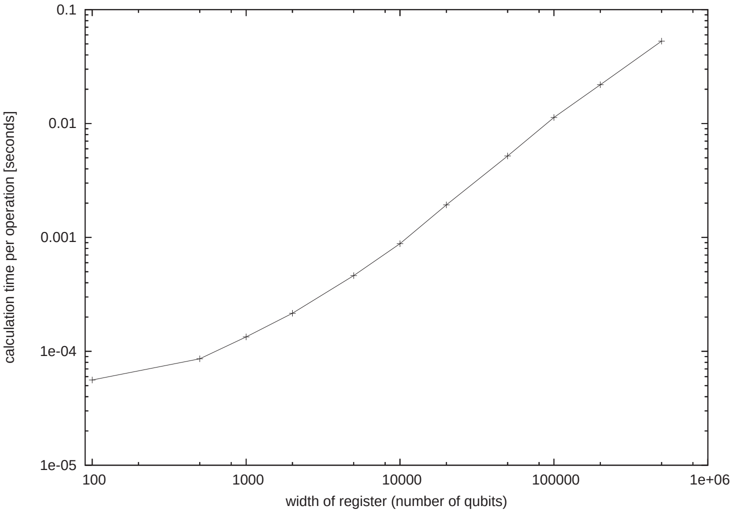


Figure 3.3: Benchmark of GraphSim for very large registers. Entanglement purification –specifically: the purification of 10-qubit cluster states with the protocol of Ref. [DAB03]– was used as sample problem. The register was filled up with cluster states to make a large ensemble, and two protocol steps were simulated. The average time per operation was obtained from the total run-time. [Giving the time per operation in seconds is of use only when one specifies the machine which has run the code: We used Linux computers with AMD Opteron processors, clocked with 2.2 GHz. Only one the machine's several processors was dedicated to our computation task. The code was compiled using the GNU C++ compiler (version 3.2.3) with 64-bit target and "O3" optimization.]

Chapter 4

Upper bounds to the fault-tolerance threshold

As already stated in Chapter 3, a Clifford network simulator such as GraphSim promises to be a useful tool in investigation fault-tolerance of quantum computation. In this chapter, I wish to outline a suggestion how this might be done. After reviewing certain relevant aspects of the theories of quantum error correction and fault-tolerant quantum computation (Secs. 4.1 and 4.2) we study how quantum error correction was simulated so far (Sec. 4.3) and then suggest a way how a stabilizer simulator might be used to improve on that (Sec. 4.4).

4.1 Brief overview on quantum error correction

When quantum computing was first proposed, it was met with substantial scepticism due to the problem of decoherence: on the one hand, a quantum register has to be shielded against uncontrolled interactions with the environment which lead to decoherence, on the other hand, one needs to interact with the quantum register in order to carry out the operations and measurements that constitute the computation. It was believed that these two requirements are impossible to reconcile as was pointed out especially by Landauer and by Unruh [Unr95]. On a classical computer, the problem is solved by storing the information in a redundant way.[1] Due to the no-cloning theorem [WZ82], one cannot make a “back-up copy” of a qubit. One can, however, distribute the information of a single (“logical”) qubit over several physical qubits such that, when a noise event of limited extent disturbs some of these

---

[1] Actually, due to the robustness of contemporary electronics, the elaborate theory of classical error correction by means of codes with redundancy is not even really needed for the volatile memory of today’s computers. Nevertheless, a tremendous redundancy is achieved by the simple fact that in a DRAM or an SRAM, a charge of not a single electron, but of a mesoscopic amount of electrons represents every single bit.

qubits, one can identify the action of the noise and undo it without measuring or disturbing the information of the logical qubit, which hence stays intact. This was realized by Shor [Sho95], who proposed what is now known as the Shor 9-qubit code. This code, however, allows only for prolonged storage of quantum information, but not for manipulating the information in order to do quantum computation without first uncoding it.

What was needed is a code which allows to perform quantum gates in an "encoded" fashion, i. e. the gates should act on the encoded qubits and produce their result in encoded form, such that the information is never "bare" and hence exposed to full decoherence. Such codes, together with schemes for the encoded gates, have been found more or less simultaneously by Calderbank and Shor [CS96] and by Steane [Ste96b] and are hence known as CSS codes. A better understanding of these and an extension to more codes was reached soon by the introduction of the stabilizer formalism [Got96, Got97]. A way to understand such stabilizer codes in terms of graphs (in a manner very similar to the graph states mentioned in the previous chapters) has also been found [GKR02, Sch01, SW02] but will not be used here.

Assuming the reader's familiarity with the subject (A good overview is provided in Nielsen and Chuang's textbook [NC00], and in Preskill's overview articles [Pre98c, Pre98a]), we only briefly review some key features which are relevant for this chapter's discussion.

The most remarkable property of CSS codes is the fact that Clifford gates are "transversal" (or can at least be decomposed into transversal gates). This means that in order to perform an encoded Clifford gate on encoded (logical) qubits, one simply acts with bare versions of the gate on of the bare (physical) qubits. To perform, for instance, a CNOT gate on two logical qubits, each encoded with 7 physical qubits according to the $[\![7,3,1]\!]$ CSS code, one performs a CNOT on each of the 7 pairs of corresponding physical qubits. After each gate, the error syndrome is measured, i.e., it is checked whether an error has occurred, and if so, which one. The syndrome can indicate for each of the bare qubits whether a bit flip, a phase flip, a bit-and-phase flip or no error has happened, provided not more errors have happened than the code is able to correct.[2]

As the Clifford gates alone do not allow for universal quantum computation, we need at least one additional gate to have a universal gate set and this gate cannot be expected to allow for transversal encoded implementation [ZCC07]. Finding a non-transversal way of implementing such a gate's action on the encoded qubits is the main challenge. The first works gave explicit construction for the chosen code either for the Toffoli gate [Sho96] or for the preparation of a specific encoded state, which in turn allows to perform a Toffoli gate [KLZ96, KLZ98b]. Later, Gottesman and Chuang introduced the

---

[2] Recall that an $[\![n,k,d]\!]$ code codes k logical qubits with n physical qubits and can detect up to d errors but only indicate the proper correction operation if at most $(d-1)/2$ qubits are affected by errors.

concept of gate teleportation which gives rise to a general scheme to implement fault-tolerant non-transversal gates with all CSS codes [GC99].

4.2 The fault-tolerance threshold

In the newer literature, a distinction is often made between the terms *quantum error correction* and *fault-tolerant quantum computation*. Subject of the former is the study of quantum codes and their error correcting capabilities, subject of the latter is the question whether this really keeps the quantum computation safe from decoherence and which additional steps have to be taken to ensure this. The goal is to construct a quantum computer capable of performing computations of arbitrary length. Just using a code as described above is not sufficient here.

Code concatenation [KL96] is a solution.[3] Leaving out certain non-negligible subtleties for a first simplified explanation, it works as follows: Given a *bare error rate* ε_0, which is the probability that an error occurs within one computational step (i. e., one gate), the probability *per step* that decoherence corrupts the state of the quantum register is reduced from ε_0 in the case of computation without encoding to $c\varepsilon_0^2$ (where c is a constant) when employing a quantum code that fails only when at least two errors occur in the same logical qubit. If one replaces each of the bare qubits again by an encoded logical qubit, that is, one concatenates the code with itself, the error rate drops to $c(c\varepsilon_0^2)^2$, and for Λ concatenation levels, one gets $\varepsilon_\Lambda = (c\varepsilon_0)^{(2^\Lambda)}/c$. It is easy to see that for $\Lambda \to \infty$ this double exponential converges *very* quickly to either 0 (the computation virtually never crashes[4]) or to 1 (it virtually always crashes), depending on whether the bare error rate ε_0 is below or above a threshold $\varepsilon_{\mathrm{th}}$. This remarkable fact is known as the *threshold theorem* and proofs of varying rigor and varying generality with respect to assumptions on the noise models have been given by various groups in 1997 [KLZ98a, KLZ98b, AB97, Kit97, Got98b]. (For a self-consistent exposition see e. g. Aharonov's PhD thesis [Aha99].) A key point is that the resource requirements only *seem* to scale exponential: At coding level Λ, one needs k^Λ physical qubits per logical qubits if the code needs k bare qubits per encoded qubit. However, as the error rate ε_Λ drops with doubly-exponential speed, one needs only a very small coding level Λ for very long computations.[5] It is easily shown that the number of physical

---

[3] Another possibility to allow for computations of arbitrary length is to use a code whose structure is invariant under the number of physical qubits used to encode a logical one and that hence can be blown up to arbitrary size. This is the feature that makes topological codes [Kit03] useful.

[4] In the parlance of fault-tolerance, an error which is not corrected and hence causes the computation to finish with a wrong result is termed a "crash".

[5] If the bare error rate is *well* (say, one order of magnitude) below the threshold, two or three levels may be sufficient. [J. Taylor, pers. comm., see also Steane's simulations mentioned later]

qubits and the number of gates is blown up only poly-logarithmically.

The brief exposition just given has been substantially oversimplified. If one wants to get a useful estimate for the value of the fault-tolerance threshold, a multitude of factors must be taken into account. Let us list a few of these complications:

- There is not just one physical error rate ε_0, but several. At the very minimum, one should distinguish: (i) the probability that performing a physical gate causes an error (gate errors), (ii) the probability that a qubit not affected by a gate suffers an error during the time step corresponding to a gate performed somewhere else (storage error), and (iii) the probability that the result of the measurement of a physical qubit is invalid (measurement errors). Furthermore, for a given physical implementation of the quantum computer, the gate error rates can be very different depending on the arity of the gate (i. e., the number of qubits it acts on) or even on the type of gate. (In most implementations, single-qubit rotation around some axes are easier to perform than others.)
- Some architectures have only limited capability to perform several gates simultaneously. (In the Cirac–Zoller ion trap proposal [CZ95], for example, only one gate per time step is possible.[6]) This makes it difficult to establish the temporal order of all the physical gates and from this estimate the typical time that a qubit has to wait for syndrome measurement and error recovery.
- With several concatenation levels, and due to the high parallelity required to keep storage errors low, the task of controlling the quantum computer becomes a very demanding task, and the capabilities of the (classical) control electronics may become a limiting factor. For instance, in semiconductor implementations, which have an especially high clock frequency (i. e., very fast gates and short decoherence times), it seems necessary to intersperse localized classical microprocessing units between groups of qubits to control operations on the spot, as already just the runtime of control signals makes control by a central classical controller impossible [J. Taylor, pers. comm., see also [TED+05]].
- In many architectures (e. g., in ion traps), a measurement takes several orders of magnitudes more time than a gate. This makes timing even more difficult.
- While two-qubit gates on low coding levels may be simulated straightforward, gates on high concatenation levels usually involve qubits far apart. In virtually all proposals for quantum computers, the time to perform such a gate depends on the distance. (Even the former standard example for an architecture with distance-independent two-qubit gates,

---

[6] On the other hand, a single ion trap is insufficient anyway, as the number of ions per trap is limited, and a "shuttling" scheme [KMW02] is needed. Then, each ion trap can work autonomously, but gates spanning two traps (which occur in high-level corrections) require expensive shuttling.

the Cirac–Zoller ion trap, suffers from this do to the need of shuttling (see footnote at item above).)

In order to get a value for the fault-tolerant threshold one has to fix the choice of architecture, setup, quantum code, recovery and parallelization schemes, a noise model and further assumptions. Then, one may follow two ways: either an analytic or semi-analytic calculation, or a simulation.

One sets out to do a rigorous calculation by making pessimistic choices for all assumption in order to get a defensible result. Everything that cannot be calculated exactly by analytical or numerical means is simplified in a way that is sure to only lower the value of the result. This will yield a *lower bound* on the threshold. The problem is that the threshold may be several orders of magnitude away from the real value. After fault-tolerance schemes were found in 1997, most researchers have first done calculations that were conservative in the described sense and should hence be considered as lower bounds. Knill et al. gave a threshold of $3 \cdot 10^{-6}$ for the $[\![7,1,3]\!]$ code, assuming independent errors and a very general architecture [KLZ98a] (elaborated in [KLZ98b]), and Kitaev [Kit97] and Aharonov and Ben-Or [AB97, AB99] also get results around 10^{-6}.

4.3 Threshold estimation by simulation

The value of 10^{-6} sounded discouraging, and it was quickly realized that it is overly pessimistic due to the fact that these calculations assume a syndrome measurement on every logical qubit on every coding level after every step. Especially, every logical qubit is treated in the same way no matter whether a computational gate has acted on it or not. In the latter case, however, only the storage error probability applies, which is likely to be much smaller than the gate error probability. The cumulative effect of the back-action of the many gates involved in syndrome measurement may then do more harm than good on these qubits, and one fares better by measuring the syndrome of "waiting" qubits less frequently. Furthermore, doing error correction always on all levels might be counter-productive as well, as only the lower levels accumulate errors fast. Finally, semi-analytic calculations usually assume concatenation ad infinitum, which is neither realistic nor gives any hint on the actual resource scaling for a bare error rate well below the threshold, which is an information as important as the threshold itself in order to chose a fault-tolerance scheme (a point stressed especially in [Ste03]).

Hence, in order to get realistic values, it is important to take the actual mode of operation (including all the point mentioned above and probably even more) into account. This renders any explicit calculation of the threshold, analytical or numerical, hardly feasible.[7] One has to resort to simula-

---

[7] Steane managed to get analytical results [Ste98a], which did take into account some more constraints but still resulted in a lower bond rather than an estimate.

tion. A first attempt to do so resulted in an estimate of 10^{-4} for the threshold [Zal96], obtained, however, with rather optimistic assumption such as negligence of storage errors and very simple treatment of two-qubit gates. A more realistic simulation proved to be quite difficult, but was finally achieved in an impressive work by Steane [Ste03], where he obtains numerical values for the threshold for the gate error as function of the storage error rate and the time required for measurements for the CSS codes $[\![7,1,3]\!]$ ("Hamming") and $[\![23,1,7]\!]$ ("Golay"). He reaches the order of 10^{-3} for the threshold, i. e., a higher value than Zalka's earlier result despite his more realistic assumptions (however still neglecting distance dependency of gates), primarily due to improved syndrome extraction procedures. This last point shows the other value of simulations: it allows to "experiment", to try out and evaluate improvements to schemes. Steane cites as his main improvement the use of the Golay code instead of the Hamming code and an improved ancilla verification scheme.

Since Steane's work, only a few other detailed numerical simulations for threshold estimation have been undertaken. Salas and Sanz studied the effect of ancilla preparation quality and of parallelisation schemes [SS02, SS04], using a technique similar to Steane's. Svore et al. simplified Steane's scheme to an only semi-numerical approach and studied how the threshold is modified due to the increasing distance between operand qubits when encoding blows up the quantum register [STD05]. The problem of ancilla verification seems crucial and the optimal solution may not yet have been found, as improvements are still being suggested (e.g., very recently: [DA07]).

Possibly a major step forward was Knill's work [Kni05] where he combines several recent ideas, most importantly ancilla factories for gate teleportation [ZLC00] and post-selection to a very sophisticated scheme for fault tolerance, which allows, according to his numerical simulations, for a fault threshold of the order of 10^{-2}. (A recent —calculating, not simulating— study [AGP07] for this scenario again has a sizable gap between the lower bound it proves, namely $1.04 \cdot 10^{-4}$, and Knill's estimate just stated.) While Knill's calculation assumes distance-independent two-qubit gates, the work by Raußendorf and Harrington claims to reach the same order of magnitude for a local architecture using very different techniques, most notably topological codes [RH07].

4.3.1 Error tracking

According to what we know, it seems highly unlikely that a quantum computer can be simulated faithfully on a classical computer (at least not without waiting exponentially long for a result). So, how can we nevertheless study fault tolerance by simulation? The technique employed by Steane [Ste03] is to track not the computational state of the computer but only the error state. This is roughly done as follows: We set out to simulate a sequence of logical

(i. e., computational) gates in a given level of encoding. The gate on top level is broken down into gates of the lower level, either transversally or according to the construction for the encoded non-Clifford gate. This is tracked down to the level of physical gates acting on physical qubits. According to the chosen scheme of fault tolerance, at any level all or some of the gates have to be followed by a syndrome measurement. The syndrome measurements require fresh ancillae, also prepared to the respective encoding level. The operations needed for this are broken down to the level of physical gates as well. Whenever a physical gate is now simulated, its action on the physical qubits is not actually evaluated, as we cannot track the actual quantum state of the simulated quantum computer. The only thing done is to simulate an error in a Monte-Carlo manner: with the given physical gate error probability, an error happens, and if so, one of the possible Pauli errors is chosen at random, and the physical qubit (or rather its representation in the simulating classical computer) is marked with the chosen Pauli operator as being erroneous. It is these error marks, and not the actual state, that the simulation keeps track of. After each gate, all the qubits not effected by the gate may also be marked as erroneous, but this with the lower storage error probability. When a multi-qubit gate acts on an erroneous gate, the error is *propagated*, i. e., the other operant qubit is marked as erroneous as well. According to the gate and its action on Pauli operators under conjugation, the type of error may change in the process.

Whenever measurements of physical qubits in the context of syndrome extraction operations are simulated, the actual result can be found despite the fact that the state of the qubits is unknown. This is because for the usual schemes, the syndrome is 0 in case of no errors, and hence, the measurement result is assumed to be 0 in case of no error or an irrelevant error (for instance, an X measurement on a qubit marked with a Z error still gives 0) and 1 for a relevant error. (If one wishes to account for measurement errors, too, one may flip the result with a given probability.) From the syndrome, the appropriate recovery operation is determined and applied. If, for example, it was determined that a certain qubit needs to be phase-flipped (i. e., a Z operation has to be applied) and the qubit was in fact marked as Z erroneous, this error mark is cleared. If it was not marked as erroneous, the recovery introduces an error: The qubit is now marked Z-erroneous.

In this fashion, one can keep track of the appearance, propagation and clearing not only of bare errors but also of errors on the encoded levels. On the low levels, faulty or insufficient recovery operations may be common, but on the highest level, they must always lead to an error-free logical qubit. If an error is propagated to the highest level and not corrected, the simulated quantum computer is deemed “crashed”. By performing many simulations and keeping track of the frequency of crashes, one obtains the crash probability, which (for a sufficiently long simulated computation) may be expected to be either close to 0 or close to 1, depending on the bare error probability. The

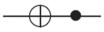
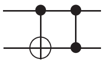
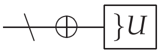
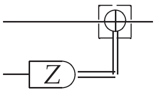
	a bit flip (NOT gate, X) and a phase flip (Z)
	a CNOT (ΛX) and a controlled phase flip (ΛZ)
	The backslash indicates that the line represents a bundle of 7 qubits. Every gate is to be thought as divided into 7 copies, each acting on one qubit. If a gate is not divided, but acts jointly on all qubits as 7-qubit gate, this is indicted by a curly brace
	a measurement and a classically conditioned gate (it is only carried out if the measurement outcome was 1)

Table 4.1: Legend to the symbols used in the quantum circuits shown in this chapter. (All circuits drawn with the LaTeX package *Q-circuit* [FE04].)

threshold is then obtained as the value of the bare error probability, for which the the simulation result changes, presumably rather abruptly, from close to 0 to close to 1.

4.4 Threshold estimation with Clifford simulations

4.4.1 Rationale

Numerical calculations, especially when based on involved algorithms, are much harder to check independently than analytical reasoning. For example, the results of the simulations cited so far have been met with scepticism in the community due to a perceived lack of opportunity for double-checking.

In this chapter I would like to propose a way to simulate fault-tolerant quantum computing in a fashion different from the error-tracking scheme just described. While the proposal might be not as fast as error tracking, it may offer useful advantages in reliability, as it allows to check certain reasoning that so far has been hard-coded into the simulation and was hence not subject of the simulation's scrutiny. Also, the same feature allows greater flexibility in trying out different fault-tolerance schemes without any need to change the simulation code.

For simplicity, I shall use for this exposition the original schemes for the $[\![7,1,3]\!]$ CSS code described in [Ste96b, CS96] in the form explained in Preskill's reviews [Pre98a, Pre98c], even though more efficient schemes are known by now. Fig. 4.1 shows the network for syndrome correction and recovery from error that should be applied to any encoded qubit after it has been exposed to noise, i.e., after a computational gate has been performed involving this qubit, or after the qubit has been untouched for a while. To read the figure, consult the legend in Table 4.1. The circuit works as follows: A logical 0 is encoded on a set of 7 bare ancillary qubits, which are coupled

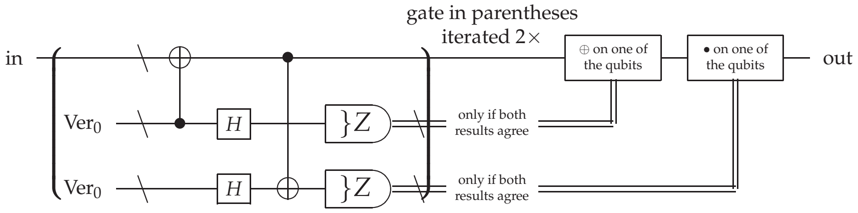


Figure 4.1: Steane's recovery scheme for CSS codes. ("Ver$_0$" means an ancillary 0, freshly prepared and verfied with the circuit of Fig. 4.2.)

by a CNOT gate, followed by a Hadamard gate. (Recall that these are both transversal gates.) According to the construction of the $[\![7,1,3]\!]$ code, each of the 7 bare qubits of the ancilla should be in the $|0\rangle$ state afterwards. If one of the 7 parallel Z measurements yields a 1, this indicates that the corresponding bare qubit in the encoded data qubit has suffered a bit flip (X) error, and as remedy, an X operation is performed on it. Unfortunately, the error might have been introduced into the ancilla by the CNOT or the Hadamard gate, so that this correction actually introduces an error instead of correcting one. This source of additional errors is suppressed quadratically by requiring the 1 to be measured twice on two independently prepared ancilla $|0\rangle$ states. In the same way, two encoded $|0\rangle$ ancillae are prepared, first subjected to a Hadamard and then coupled to the data with a CNOT (now in opposite direction) to detect and correct for phase flip (Z) errors.

Even with this double checking, we still cannot expect that the correction extracts more noise than it adds. This is because each of the ancillae may contain errors which are propagated onto the data qubit due to the back-action of the CNOT gates. This possibility has to be suppressed quadratically as well by verifying *twice* that the ancilla is really in the $|0\rangle$ state before allowing it to interact with the data. Fig. 4.2 shows a standard version of the preparation and verification network (still following [Pre98a, Pre98c]).

Can we now be sure that this scheme removes errors which appear with a rate ε_0 but introduces new errors only with a rate of $O(\varepsilon_0^2)$, as needed to make code concatenation work? Yes: the networks can be rigorously checked by paper-and-pencil calculation, as explained in the cited references. But what is the exact probability that a qubit that initially was erroneous with probability ε_0 is erroneous after the circuit? How does this depend precisely on the, possibly different, error rates for the various gates? How on the preparation and the measurement noise? As argued earlier, the order in which the operations are performed, may make a huge difference unless we neglect storage errors. Getting a closed formula for this might be still possible, but is surely very laborious and extremely error-prone. If we want to include interactions between different coding levels, things get only more difficult. So we better resort to a numerical *simulation*: After every time step, we draw for each qubit

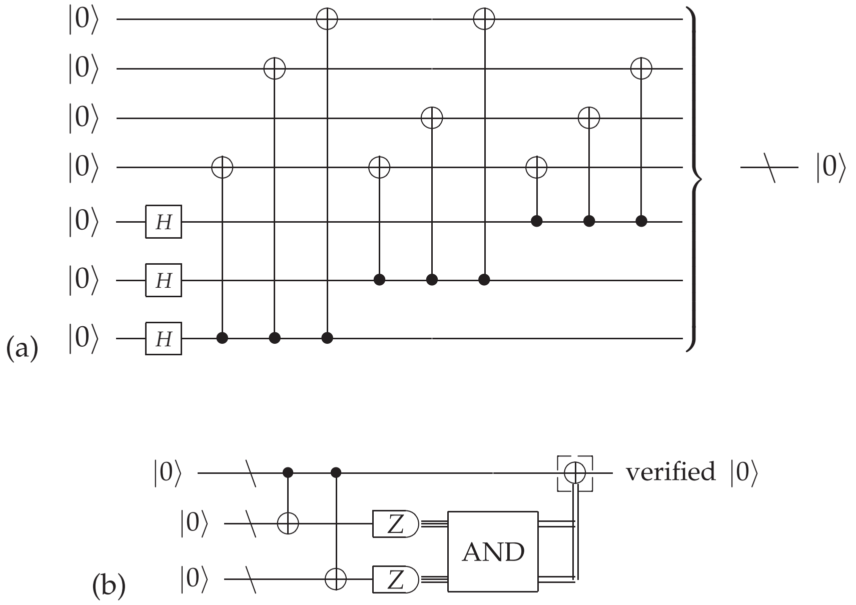


Figure 4.2: (a) Preparation of an encoded $|0\rangle$. (b) Verification of the encoded ancilla (using up two further encoded zeros) for use as ancilla for Steane recovery.

a number from a pseudo-random number generator (with range $[0;1]$), and if the number is below the error rate (which depends on whether the qubit was idle or participated in a gate, and in the latter case also in the kind of gate), one of the "error operators" X, Y and Z is applied to it.[8]

This is the way used by Steane to simulate noise. As described above, in Steane's scheme, the path of the error is then "tracked". For this tracking, one needs to supplement expectations about where the errors become manifest (measured) and where they get corrected. This information comes from an understanding of the functioning of the network, namely when which conditioned correction operations happen and in which cases they actually succeed in correcting the error. In other word, the reasoning, why and how the correction network operates, has to be put into the simulation by the programmer. The program does not notice by itself that a certain measurement is a syndrome measurement, or a certain operation clears an error.

It would be desirable if the correcting behaviour emerged from the simulation, i. e., if the error correction happens "automatically" simply because an error-correction network is simulated. This should be most useful if one

---

[8] After two-qubit gates, we may chose from one of the 15 possibilities in $\{\mathbb{1}, X, Y, Z\} \otimes \{\mathbb{1}, X, Y, Z\} \backslash \mathbb{1}^{\otimes 2}$. This adds the possibility of multi-qubits gates introducing *correlated* errors in a most straight-forward way.

uses complicated schemes or wishes to study effects of code subtleties[9], and would free the programmer from the very difficult and error-prone task of putting in the recovery logic "manually". Furthermore, an automatic treatment of recovery logic may save considerable time for a researcher who wishes to compare the performance of different codes and recovery circuits under varying constraints and parameters for noise, geometry and other factors, or who even seeks to find new, or improved recovery networks. On first look, such an automatism seems not possible because it amounts to simulating the actual action of the gates involved in the error correction. Yet, as these gates are all Clifford gates, a simulation with the help of a stabilizer simulator should be possible.[10] The main problem is that while the gates of the syndrome measurement and the error correction parts of the network are all Clifford gates, the computational gates typically are not and hence drive the simulated computer out of the space of stabilizer states. In the following section we look at the simple case of a computational network only consisting of Clifford gates, and then, we shall study a simple scheme to deal with non-Clifford gates and argue that it may be expected to give reasonably accurate results despite its simplicity.

4.4.2 Simulation of pure Clifford networks

As long as the computational network (i. e., the network of gates acting onto the actual data as represented by the top coding level) consists only of Clifford gates, the whole physical network will also only consider of physical Clifford gates and can be fully simulated on a classical computer. This is because, at least for CSS codes, the circuits for encoding, syndrome extraction, and recovery can always be build using only Clifford gates, and because the computational Clifford gates can be broken down into transversally applied physical Clifford gates. The full physical circuit is then simulated, with errors being put in at random with specified probabilities in order to do a Monte-Carlo style simulation. At the very end of each run, the result is measured, decoded and compared to the expected result.[11]

Fig. 4.3 shows such the results for such a simulation for a quite simple computational network: A logical qubit is initialized as $|0\rangle$ (according

---

[9] To give a well-known example for such a subtlety: The $[\![7,1,3]\!]$ CSS code is usually said to be able to correct but one error. Yet, if two *different* bare qubits are affected by different errors —one qubit by an X error, the other one by a Z error— the network of Fig. 4.1 can recover the state.

[10] I may be not the first to suggest such an approach. Knill may have used similar techniques to obtain the results presented in [Kni05], as seems to be implied in the supplemental material to the article. One should, however, expect that the scaling of simulation techniques following Aaronson and Gottesman [AG04] may have been an obstacle. This limitation was, after all, the motivation to develop GraphSim.

[11] Note that in case of a complicated computational circuit, the Clifford simulator (without encoding, and with noise set to 0) can also be used to find out what result to expect.

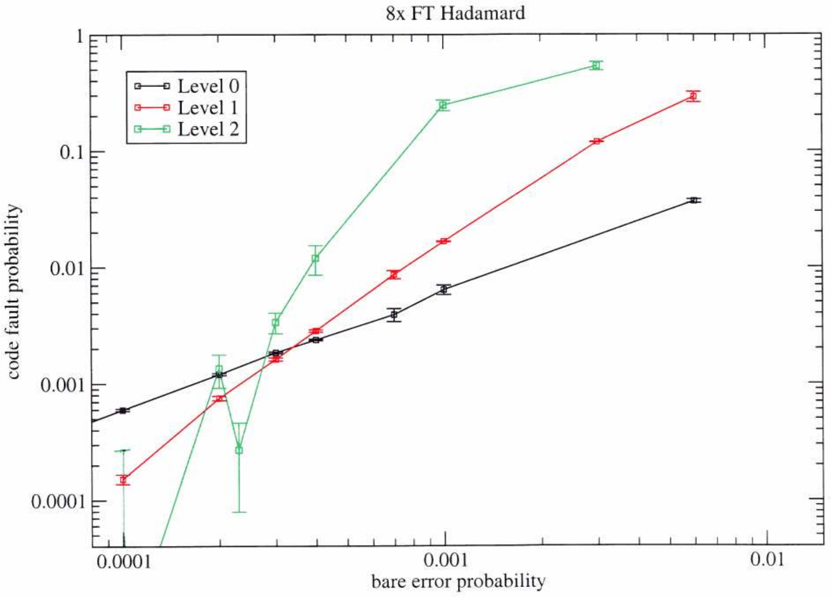


Figure 4.3: Simulation of fault-tolerant performance of 8 consecutive Hadamard gates at different bare error rate and coding levels. The curves seem to cross roughly at a bare error probability a bit less than $3 \cdot 10^{-4}$. (For a doubly-concatenated code (green curve) the calculation gets expensive and hence, I have made considerably less Monte Carlo runs for this first try, resulting in a much higher statistical inaccuracy. This may have caused the outlier to the left-hand side — note that the error bars denote only one standard deviation and an error larger than indicated is likely.)

to Fig. 4.2a), and then, 8 consecutive encoded transversal Hadamard gates are applied, each gate followed by a full recovery network according to Fig. 4.1. Afterwards, a logical Z measurement is simulated. If the result (after post-measurement error correction) is not 0, the computation is considered as crashed. The plot shows this fault probability for different values of the bare error probability ε_0. After each physical gate (including the gates within the recovery circuits, of course), an error event happens with probability ε_0. In case of a single-qubit gate, an extra X, Y or Z operation is then applied, each with probability $1/3$. For a two-qubit gate, one of the 15 possible errors in $\{\mathbb{1}, X, Y, Z\}^{\otimes 2} \backslash \mathbb{1}^{\otimes 2}$ is chosen uniformly at random. Each measurement reports, with the same probability, the wrong result. Storage errors have been set to 0 for this simple example.

The black curve shows the fault probability for an non-encoded circuit, i. e., for simply 8 consecutive bare Hadamard gates applied to a qubit initially in the $|0\rangle$ state, followed by a single Z measurements. Disregarding the possibility of two errors canceling each other, we get a fault probability of roughly ε_0^{10}, as there are 10 operations. In fact, the doubly logarithmic plot shows a linear curve, indicating a simple power-law dependence. The red curve is done for a single code layer (i.e., 7 bare qubits, no code concatenation). We see that for a bare error probability ε_0 below approx. $3 \cdot 10^{-4}$, the coding and error correction succeeds in lowering the fault probability, while above this threshold, the error recovery network introduces more noise than it corrects for. The green curve is for a two-layer code (i.e., $7^2 = 49$ qubits, code concatenated once). The green curve should be expected to cross the other two curves roughly at the same point.[12] Within the specified 1σ error bars, this seems to be roughly the case. So far, the calculation seems to reproduce the initially quoted rough results of an fault tolerance threshold slightly above 10^{-4} for a straight-forward implementation of the concatenated $[\![7,1,3]\!]$, neglecting storage errors with syndrome measurements on all levels after every step.

4.4.3 Generation of encoded circuits

Even the simple eight-Hadamards network described above blows up to considerable size when encoded at two levels. Together with the gates of the recovery networks on both coding levels (and with the lower-level recovery circuits for each gate in the higher level recovery circuits), we get a bit more then 10^3 physical gates. Obviously, such a network is too cumbersome to implement manually; one should generate it by a computer program. This can be done in an elegant fashion by modeling code concatenation and error correction with object-oriented design. (A reader not familiar with the terminology of object-oriented programming may want to skip to the next

---

[12] Only roughly, not exactly, due to subtle "interactions" (such as error cancellation) between code layers.

subsection.) An abstract class `qubit` is used to represent a qubit and has two concrete subclasses, `phys_qubit` for a physical (bare) qubit and `enc_qubit` for an encoded qubit. An `enc_qubit` has an array field `sub_qubits` with references to the lower-level `qubits` it is comprised of, while a `phys_qubit` simply knows its index (i. e., position) within the quantum register simulated by GraphSim. For all commonly used transversal Clifford gates, `qubit` provides an abstract method. In `phys_qubit`'s implementation of this method, the corresponding method of GraphSim's `GraphRegister` is called, and then, a random choice is made with the given bare error probability to decide, whether an error event happens. In the error case, the chosen error event is realized by calling GraphSim's methods for Pauli operations on the affected qubits. In `enc_qubit`'s implementations for the transversal Clifford operations, the same method is called for the `sub_qubits`, which hands the gate recursively down the code layers. After calling the `sub_qubits`' methods, the method calls the `recovery` method, which executes the full recovery circuit of Fig. 4.1 by creating ancillae, performing all the gates and measurements (by calling the relevant methods of the ancilla `enc_qubits` on the same coding level) and indicated recovery operations. Measurements are implemented in the same fashion.

As the `recovery` method creates ancillae and then disposes of them, it is useful to implement a simple resource management for the quantum register. This is most conveniently done in the style of a primitive heap: When an encoded $|0\rangle$ ancilla is needed, the necessary physical qubits are "allocated", and once the ancilla has been measured, they are "freed". The heap manager just keeps track of which physical qubits are not in use ("free"). For an allocation, a free qubit is chosen arbitrarily, marked as "not free", set to $|0\rangle$ and provided. When it is freed, it is simply marked "free" again. Note that this simple scheme is viable only if one simulates a quantum computer whose performance at multi-qubit gates does not depend on the distance of the operand qubits, i. e., there are no geometry considerations. However, all proposed implementation do have geometry dependences, and the fault-tolerance threshold of a given scheme can depend strongly on the strategy to allocate ancilla qubits. (See e. g. [Ste02] for a discussion on geometry optimization.) Hence, the simple scheme just sketched may be replaced by various complex schemes, which allows to compare their performance. Such simulations may turn out to be of high importance, once one considers detailed quantum computer designs, and the possibility that they can be studied with a Clifford simulator in the proposed manner might then turn out to be most valuable.

4.4.4 Notes on Monte Carlo statistics

So far, we have considered the naïve strategy of drawing after each physical gate a (pseudo-)random number from $[0;1]$, and if it is below ε_0, simulate a Pauli error. However, as can be seen from Fig. 4.3, the values of the bare error probability ε_0 for that the simulation needs to be performed, can be quite low. Hence, even for a larger number L of gates (for the level-2 curve in the 8-Hadamard example, $L > 1000$), the probability that all these gates perform without any bare error, namely $(1-\varepsilon_0)^L$, is quite high. For example, for $\varepsilon_0 = 10^{-4}$ and $L = 10^4$, 90% of all simulation runs will not contain a single error event, and hence are identical. This shows that this approach is very inefficient.

A better solution uses the binomial distribution and works as follows: Let us assume that the computation contains at most L gates. (Note that the number of gates depends on the occurring errors, as these stimulate the insertion of gates for error recovery according to the measured syndrome. Hence, we should choose L so large that the probability of the actual number of gates exceeding L is so small that these cases may safely be neglected.) Then, the probability of k errors happening within these L possible "error locations" is given by

$$p(k) = \binom{L}{k} \varepsilon_0^k (1-\varepsilon_0)^{L-k}.$$

Not all of the k errors actually occur within a computation as some may be beyond the actual number of gates. We now let k take a sequence of values, keeping the value fixed for a number of Monte Carlo runs. For every run, we choose, uniformly at random, k error locations from the set $\{1, 2, \ldots, L\}$ of error locations. Within the run, each gate increases a global counter and then reads of the counter to see, which position it (the gate) has within the simulation sequence of all physical gates. Instead of drawing a random number from $[0;1]$, the gate looks up whether its position (as read off from the counter) is one of the k error locations drawn at the start of the run, and if so, simulates a Pauli error. Let us say that of the $n(k)$ Monte Carlo runs simulated for a given value of k, a certain number $\nu(k)$ resulted in a fault, i. e., a wrong end result. If the actual probability that a fault occurs for k chosen error locations is $f(k)$, then $\nu(k)/n(k)$ is an unbiased estimator for $f(k)$, and the uncertainty of this estimation is given by $\sqrt{n(k)f(k)(1-f(k))}$. (This is because the probability that within the $n(k)$ runs one finds $\nu(k)$ faults is given by $\binom{n(k)}{\nu(k)} f(k)^{\nu(k)}(1-f(k))^{n(k)-\nu(k)}$. This binomial distribution has a variance of $n(k)f(k)(1-f(k))$).

This then allows to estimate the probability of a fault under independent errors as a sum of the frequencies $n(k)/\nu(k)$ weighted by the binomial probabilities $p(k)$:

$$P_{\text{fault}}(\varepsilon_0) = \sum_{k=0}^{L} p(k) \frac{\nu(k)}{n(k)}. \tag{4.1}$$

The uncertainty of this estimation is at the 1σ level

$$\Delta P_{\text{fault}}(\varepsilon_0) = \left[\sum_{k=0}^{L} p^2(k) n(k) \frac{\nu(k)}{n(k)} \left(1 - \frac{\nu(k)}{n(k)}\right)\right]^{1/2} \tag{4.2}$$

It is easy to see that many of the terms in Eq. (4.1) are negligible. Choosing a cut-off k_{cut}, we may argue that for $k > k_{\text{cut}}$, $p(k)$ is so small that P_{fault} does not change significantly if we replace all the frequencies $\nu(k)/n(k)$ with $k > k_{\text{cut}}$ by $\nu(k_{\text{cut}})/n(k_{\text{cut}})$, i.e.

$$P_{\text{fault}}(\varepsilon_0) = \sum_{k=0}^{k_{\text{cut}}} p(k) \frac{\nu(k)}{n(k)} + \underbrace{\left(1 - \sum_{k=0}^{k_{\text{cut}}} p(k)\right) \frac{\nu(k_{\text{cut}})}{n(k_{\text{cut}})}} + \mathcal{O}(\varepsilon_0^{p_{\text{cut}}}). \tag{4.3}$$

The rest term $\mathcal{O}(\cdot)$ can even be bounded easily. In the stochastic limit of many Monte Carlo runs, the frequencies decrease strictly monotonically, $\nu(k+1)/n(k+1) < \nu(k)/n(k)$, and hence, the rest term is negative and its modulus is strictly smaller than the under-braced term in Eq. (4.3). The purpose of this approximation is easy to see. Once one chooses a cut-off k_{cut} large enough such that the approximation error $|\mathcal{O}(\cdot)|$ is small compared to the value of P_{fault}, one knows that it suffices to run Monte Carlo simulations only for $k < k_{\text{cut}}$. The cut-off error is simply added to the statistical error of Eq. (4.2) in order to give the full error to be denoted with error bars.

If k_{cut} has to be chosen large, one may not want to wait for all k_{cut} sets of Monte Carlo runs to have finished. A solution is then to skip some runs, i. e., to only perform runs for $k \in \{\Delta k, 2\Delta k, 3\Delta k, \dots, k_{\text{cut}}\}$. The missing frequencies $\nu(k)/n(k)$ can then be replaced by the frequencies for the next-lower value of k actually calculated, in a manner similar to that used in the previous paragraph. Again, the truncation error thus introduced can be bounded and made small by choosing Δk not too large. Note that care must be taken to adjust Eq. (4.2) to the case of $\Delta k > 1$: Variances that occur several times in the sum must not be added Pythagoreically but directly, as they are no independent sources of errors.

In practice these two truncation techniques allow for a saving in computing time by orders of magnitude without increasing the error a lot.

4.4.5 Treatment of non-Clifford gates

One may argue that the whole scheme presented so far is unable to provide realistic estimates for fault-tolerance thresholds because it does not allow to simulate computational networks with non-Clifford gates. This seems indeed to be a severe objection because a quantum computer without non-Clifford gates is of no use due to the Gottesman-Knill theorem and the presence of non-Clifford gates in fact does reduce the fault-tolerance threshold.

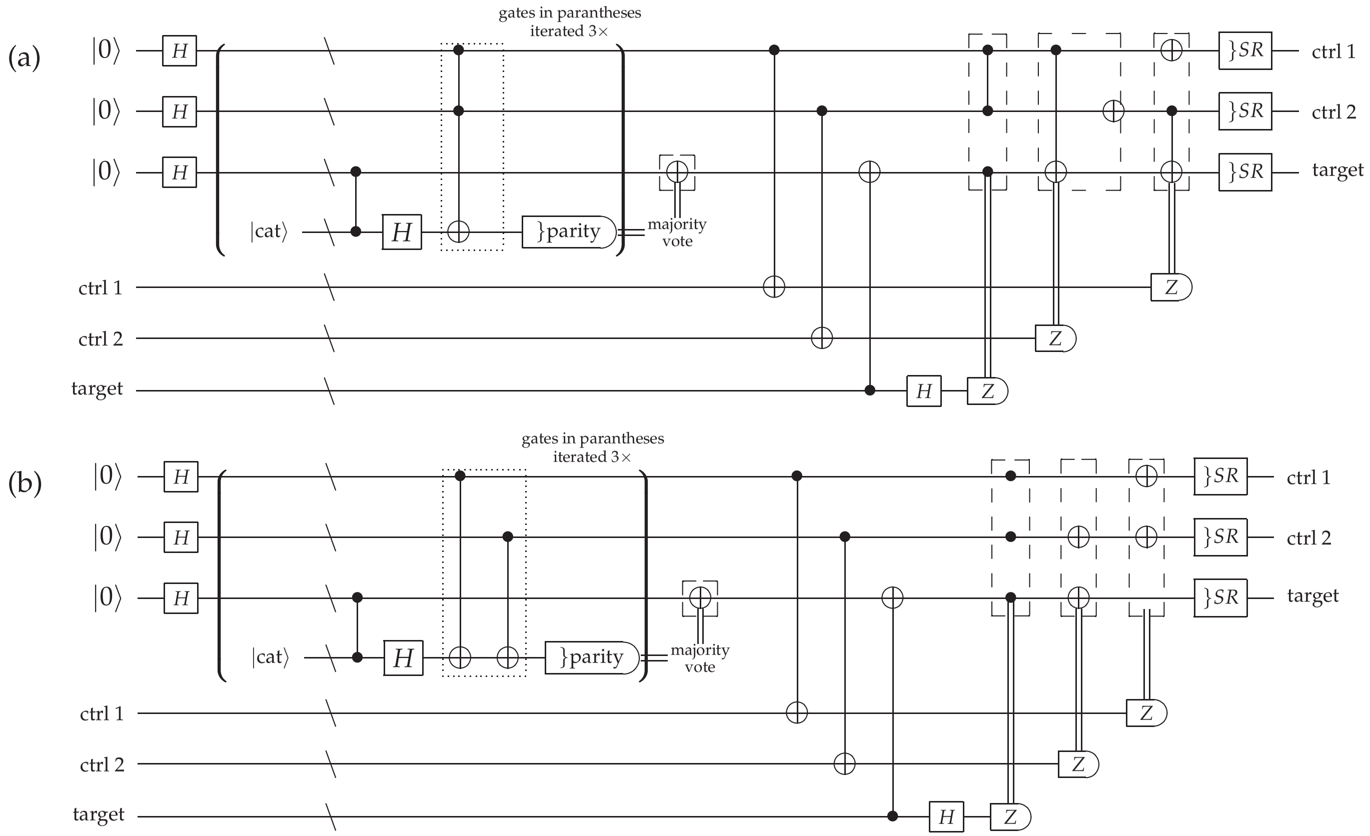


Figure 4.4: (a) Shor's fault-tolerant construction [Sho96] for performing the Toffoli gate in the $[\![7,3,1]\!]$ CSS code as explained in [Pre98a, Pre98c]: A "cat state" $|\text{cat}\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes 7} + |1\rangle^{\otimes 7}\right)$ is prepared and verified and then coupled to the operand qubits by transversal operations. (b) A "placebo" construction for a Clifford simulation.

The reason that the code fault probability is increased by non-Clifford gates seems to be, however, not related to the fact that they cannot be simulated efficiently on a classical computer.[13] The reason is rather that non-Clifford gates cannot be implemented transversally. For a transversal gate, the logical qubit is in the code space before the gates on the constituent qubits have been performed and again in the code space afterwards. Hence, if the sub-gates are performed simultaneously, the qubit leaves the code space only momentarily or even not at all. Hence, it always enjoys the protection of the code on the given concatenation level. In a non-transversal gate, however, the sub-gates (i. e., the gates on the level of the constituent qubits) cause the logical qubit to leave the code space until the whole procedure is finished.

Fig. 4.4a shows Shor's construction [Sho96] of a fault-tolerant Toffoli gate (which is actually an instance of the general gate-teleportation scheme found later by Gottesman and Chuang [GC99]): The ancillary "cat state"

$$|\text{cat}\rangle := \tfrac{1}{\sqrt{2}}\Big(|0\rangle^{\otimes 7} + |1\rangle^{\otimes 7}\Big)$$

is not a valid code word, and hence, the $|0\rangle$ ancillae leave the code space as soon as they are coupled to the cat state by the transversal bare CNOT and Toffoli gates in the parentheses. As we cannot guarantee any longer for their correctness by syndrome checking, the ancillae are verified by the parity measurement on the cat state, done twice as always with fault-tolerant verifications. Even though the ancillae are now verified to be correct, they are no longer in the code space and cause the operand qubits to leave the code space as well after they interact with them due to the three transversal CNOTs. Then, the operand qubits are measured, teleporting the result to the ancillae which now become data qubits. These data qubits now undergo conditioned transversal correction operations. Only after these, the logical data qubits may be expected to be in the code space again, and only then, a syndrome measurement (denoted "}SR" in the circuit) can be performed.

To summarize: In a transversal gate, a syndrome measurement can be performed immediately after each constituent qubit has been touched by just a single gate. In the Toffoli construction, however, each constituent qubit is touched by up to five gates before its syndrome can be checked for the next time, and hence, the probability that more than one error has occurred and the recovery thus fails is significantly increased.

It seems reasonable to assume that this is the by far dominant reason why non-transversal gates reduce the fault-tolerance threshold. The idea is now that the effect of this delay in syndrome measurement can be faithfully modelled by a purely Clifford network. We simulate, instead of the Toffoli construction of Fig. 4.4a, a network which is equally complex but contains only

---

[13] "is not related" might be too strong a statement. A subtle relation certainly does exist between the Gottesman-Knill theorem and the actual reason given now, but I strongly doubt that this invalidates the reasoning to be exhibited in this section.

Clifford gates. In Fig. 4.4b, such a network is shown, called a "placebo Toffoli". In the place of the bare Toffoli gate, it performs a three-qubit Clifford gate. This could be any Clifford gate; here, the one obtained from a sequence of two CNOTs has been chosen. After the simulation of this three-qubit gate (which is a sequence of two gates for the underlying Clifford simulator but considered as one gate within the fault-tolerance simulation), a noise event is simulated with the usual probability, but the Pauli operation is now chosen from $\{\mathbb{1}, X, Y, Z\}^{\otimes 3} \backslash \mathbb{1}^{\otimes 3}$. If a concatenated code is simulated, the lower-level double-CNOT is not simulated by performing transversally the CNOTs but by recursively performing the whole placebo network on the lower level (unless the lower level is the physical level, of course). As in the real network, the simulator abstains from doing any syndrome measurements after any of the transversal gates even though the logical qubits do not leave the code space. The conditional operations that finish the teleportation (marked with dashed boxes in Fig. 4.4) are different to those in the real Toffoli case. They have to be changed such that the placebo Toffoli network actually does perform the placebo Toffoli operation, i. e., the double CNOT, on the upper level. Nevertheless, after the conditional correction, one may simulate error events such as if the two-qubit correction operations of the Toffoli case had been performed and not just the simpler local operations in the placebo network. This ensures that correlated errors are introduced with the same rate as in the real Toffoli network.

I regret that I cannot present numerical results for an example computation that involves the Toffoli placebo construction. I have made preliminary tries, which however have failed to give conclusive results before we decided to freeze this project in order to first pursue another application of the GraphSim Clifford simulator, which will be described in the following chapter. However, even though I was not able to finish this work in the present thesis by providing numerical examples for simulations of a placebo construction I still feel that this is a promising ansatz which may be worth further pursuing.

4.5 Discussion

Given a set of bare error parameters, networks building blocks and a timing/parallelization scheme, can we expect that a simulation of the described kind gives an accurate estimate for the failure rate of the real quantum computation? As long as only Clifford gates are involved this should be the case, as the simulation is exact for Clifford networks. If non-Clifford gates are simulated in the fashion presented in the previous section, a sceptic may doubt that the "placebo" construction really captures all conceivable modes of failure. For example, the logical qubit leaves the code space for considerable time while the placebo construction stays in the code space. Without a rigorous proof it is not justified to claim that the failure probability of the placebo construction equals that of the real gate. However, it seems a very reasonable assumption to say that the placebo construction will not be *more* likely to fail.

This last point should be elaborated more clearly. Fault-tolerant operation requires that all gates within the chosen universal gate set can be performed fault-tolerantly, and some gates may put stricter limits onto the bare error probability than others. As before, we consider the typical case where this gate set consists of a number of Clifford gates and one non-Clifford gate, say, the Toffoli gate. Let us first assume that our quantum computer is only capable of performing Clifford gates, and that it is possible to perform such calculations of arbitrary length with arbitrarily small probability of failure if the bare error rate is below a threshold $\varepsilon_{\mathrm{th}}^{\mathrm{C}}$. The existence of such a threshold is guaranteed by the threshold theorem. If we now improve the quantum computer and give it the capability of performing the non-Clifford gate needed to make its gate set universal, this change cannot make the threshold larger. This is because the Clifford gates are still needed and also because the fault-tolerant construction for this non-Clifford gate involves Clifford gate which have to be performed fault-tolerantly. Hence, the threshold for universal quantum computation will be smaller than that for pure Clifford operation, $\varepsilon_{\mathrm{th}}^{\mathrm{U}} \leq \varepsilon_{\mathrm{th}}^{\mathrm{C}}$.

We have argued that our simulation allows us to get sound estimates with controlled error (which can, in principle, be made arbitrarily small by running enough Monte Carlo simulations) for $\varepsilon_{\mathrm{th}}^{\mathrm{C}}$. Furthermore, if we use a “placebo” construction, we may get a reasonable estimate for $\varepsilon_{\mathrm{th}}^{\mathrm{U}}$. However, the error of this estimate cannot be made arbitrarily small because we cannot exclude the possibility of systematic error which are not accounted for. Still, because of $\varepsilon_{\mathrm{th}}^{\mathrm{U}} \leq \varepsilon_{\mathrm{th}}^{\mathrm{C}}$, our estimate for $\varepsilon_{\mathrm{th}}^{\mathrm{C}}$ will be (within the margin of its statistical uncertainty, which can be made small) an *upper* bound to the true threshold $\varepsilon_{\mathrm{th}}^{\mathrm{U}}$. Furthermore, the estimate for $\varepsilon_{\mathrm{th}}^{\mathrm{U}}$ will be smaller than the estimate for $\varepsilon_{\mathrm{th}}^{\mathrm{C}}$, i. e. be a better upper bound to the true value $\varepsilon_{\mathrm{th}}^{\mathrm{U}}$. We may even expect it to be a rather tight upper bound if the agree to assume that the residual systematic errors just discussed are small.

This hope is not unreasonable, but even if one is not willing to make this assumption, we have gained something, because we need it only to claim that the simulation results qualify as *estimates* for the true threshold value. The claim that they are *upper bounds* holds even in the face of non-negligible systematic errors, because these can only cause the estimate to be too large. Virtually all published results on fault-tolerance thresholds are either strict lower bounds or uncontrolled estimates. As there is good reason to feel uneasy about uncontrolled estimates (i. e., estimates for which a statistical or systematic uncertainty cannot be states), it is useful to have a method that provides upper bounds because if these turn out to be close enough to the estimates, they help to constrain the margin of error of these estimates.

Chapter 5

PUBLICATION: Quantum communication cost of preparing multipartite entanglement

Caroline Kruszynska, Simon Anders, Wolfgang Dür, and Hans J. Briegel

published in *Physical Review A, Vol.* ***73*** *(2006), 062328*

submitted 4 Jan 2006, published 20 Jun 2006
DOI: 10.1103/PhysRevA.73.062328

ArXiv preprint: quant-ph/0512218 v2

Abstract

We study the preparation and distribution of high-fidelity multi-party entangled states via noisy channels and operations. In the particular case of GHZ and cluster states, we study different strategies using bipartite or multipartite purification protocols. The most efficient strategy depends on the target fidelity one wishes to achieve and on the quality of transmission channel and local operations. We show the existence of a crossing point beyond which the strategy making use of the purification of the state as a whole is more efficient than a strategy in which pairs are purified before they are connected to the final state. We also study the efficiency of intermediate strategies, including sequences of purification and connection. We show that a multipartite strategy is to be used if one wishes to achieve high fidelity, whereas a bipartite strategy gives a better yield for low target fidelity.

[PACS: 03.67.Mn, 03.67.Hk, 03.67.Pp]

5.1 Introduction

In the past, (multipartite) entanglement has been mainly considered as a puzzling artifact of quantum mechanics. More recently, however, the focus on entanglement has shifted, as it was realized that entanglement also constitutes a valuable resource for quantum information processing. Possible applications of multipartite entanglement include certain security tasks in distributed communication scenarios [HBB99, CL04, CGS02], the improvement of frequency standards [WBI$^+$92, WBIH94], as well as measurement based quantum computation schemes [RBB03].

In this context, the problem of generating multipartite entanglement of high fidelity arises. If entangled states are to be distributed among spatially separated parties, as it is e. g. required in distributed communication scenarios, the main obstacle comes from channel noise. Possible ways to overcome channel noise and hence to successfully generate high-fidelity multipartite entangled states have been developed. These methods are based on (i) quantum error correction and make use of (concatenated) quantum error correction codes [KL96], or (ii) entanglement purification [BBP$^+$96, DEJ$^+$96, DAB03, ADB05]. While (i) is applicable to directly distribute arbitrary states, (ii) concentrates on the generation of specific, maximally entangled pure states. The generation of maximally entangled pairs of particles allows in turn to distribute arbitrary states by means of teleportation. In both cases, a substantial overhead is required to guarantee successful, high fidelity generation of the desired states. In (i) this overhead arises from redundant encoding, enabling one to perform error correction, while for (ii) several identical copies need to be prepared and locally processed to generate high fidelity entangled states. The quantification of this overhead, or the quantum communication cost, which we shall define more precisely, is the main concern of this article.

To be specific, we will concentrate on schemes based on entanglement purification. These schemes are specially suited to generate entangled states of a specific form, and are hence expected to perform better than general purpose schemes such as (i). In fact, a remarkable robustness of entanglement purification protocols against noise in local operations –which we consider in addition to channel noise– has been found [DAB03, ADB05]. That is, errors of the order of several percent in local control operations can be tolerated, still allowing for the generation of high fidelity entangled states, even in the presence of very noisy quantum channels and with only a moderate overhead. For perfect local operations, the required overhead in resources is solely determined by the noisy quantum channels. In this case, the channel capacity [BKN00, DCH04] provides a suitable measure for this overhead. In a bipartite communication scenario, the channel capacity gives the optimal rate of quantum communication, i. e. the amount of quantum information transmitted per actual channel usage. While one might think that the abstract notion of channel capacity may also be employed to our problem –the generation of certain high fidelity entangled pure states–, one actually faces a number of difficulties. First, channel capacities are asymptotic quantities which are very complicated to calculate; second, the definition of channel capacity is not suitable to account for imperfect local operations (e. g. noise in local coding and decoding procedures); and third, we are actually considering a restricted problem, namely the generation of specific multipartite entangled states, rather than the successful transmission of arbitrary quantum information.

We thus introduce a quantity closely related to quantum channel capacity, namely the *quantum communication cost* $C_{F,G}$. $C_{F,G}$ denotes a family of quantities which specify the number of uses of the noisy quantum channel required to prepare a specific (multipartite) entangled state $|G\rangle$ with fidelity $\tilde{F} \geq F$. In this paper, we will focus on target states $|G\rangle$ which are so-called two-colourable graph states. These states include, for instance, GHZ states and cluster states –a universal resource for measurement based quantum computation [RBB03]– and they are locally equivalent to codewords of Calderbank-Shor-Steane error correcting codes [Ste96b, CS96]. We establish upper bounds on $C_{F,G}$ by optimizing over a large class of different strategies that generate these multipartite entangled states. These strategies include, as extremal cases, (i) the generation and purification of pairwise entanglement, from which, by suitable connection processes (or, alternatively, teleportation) the desired multipartite states are generated; (ii) the generation and direct multipartite purification of the desired target states. Intermediate strategies, e. g. the purification of smaller states to high fidelity and their subsequent connection to the desired larger state, will also be investigated. Depending on the actual noise parameters for channels and local control operations, and on the desired target fidelity F, the optimal strategy varies. For high target fidelities, multipartite strategies turn out to be favorable.

This article is organized as follows: In Sec. 5.2, we present the concepts we will use: We start with a review of the graph state formalism in order to introduce notation and the two types of states we wish to distribute, namely GHZ states and 1D cluster states. We shall also introduce a technique to connect two smaller graph states to obtain a larger one. Then, we give details for our noise models and review the employed purification protocols. Readers familiar with these concepts may skip this section. Sec. 5.3 explains the different strategies for employing the protocols that we wish to compare. The actual comparison is done using extensive numerical Monte Carlo simulations and results are presented in Sec. 5.4. In order to corroborate these results we have done analytical studies for certain restricted noise models (Sec. 5.5). We conclude with a summary (Sec. 5.6).

5.2 Basic concepts

5.2.1 The graph-states formalism

A graph $G = (V, E)$ is a collection $V = \{a, b, c, \dots\}$ of $N = |V|$ vertices connected by edges $E \subset [V]^2$. The description of the edges is given by the adjacency matrix Γ_G associated with the graph

$$(\Gamma_G)_{ab} = \begin{cases} 1, & \text{if a and b are connected by an edge,} \\ & \text{ i.e. } \{a,b\} \in E \\ 0, & \text{otherwise} \end{cases}$$

The neighborhood $N_a \subset V$ of vertex a is defined as the set of vertices connected with it by an edge, $N_a = \{b : \{a,b\} \in E\}$.

With each graph G we associate a pure quantum state. If the graph's vertex set can be separated into two sets A and B such that no edges exist between vertices of the same set, we call it a two-colourable graph[1]. The vertices are qubits and the edges represent interactions.

There are three equivalent descriptions of graph states which are reviewed in the following sections (For a detailed treatment see [HEB04]):

Graph states in the stabilizer formalism

Associated with a graph G is a set of N operators

$$K_G^{(a)} = \sigma_x^{(a)} \prod_{b \in N_a} \sigma_z^{(b)}. \tag{5.1}$$

They form a complete set of commuting observables for the system of qubits associated with the graph and therefore possess a set of common eigenstates which form a basis of the Hilbert space. These eigenstates are called graph states and are here written as $|G, \boldsymbol{\mu}\rangle$ where the a^{th} component of vector $\boldsymbol{\mu} \in \{0,1\}^N$ is equal to 0 if $K_G^{(a)} |G, \boldsymbol{\mu}\rangle = |G, \boldsymbol{\mu}\rangle$ and 1 if $K_G^{(a)} |G, \boldsymbol{\mu}\rangle = -|G, \boldsymbol{\mu}\rangle$. We abbreviate $|G\rangle := |G, \mathbf{0}\rangle$. We also sometimes suppress the letter "G" and write just $|\boldsymbol{\mu}\rangle$, if the context makes clear which graph G is meant.

Graph states in the interaction picture

A graph state with $\boldsymbol{\mu} = \mathbf{0}$ can be written in the computational basis in the following manner:

$$|G\rangle = \left(\prod_{\{a,b\} \in E} \Lambda Z^{(ab)} \right) |+\rangle^{\otimes |V|} \tag{5.2}$$

---

[1] Mathematical literature prefers to call such graphs *bipartite* but we reserve this term to denote operations or settings comprised of two different sites (locations).

where ΛZ is the controlled phase gate,

$$\Lambda Z^{(a,b)} = |00\rangle_{ab}\langle 00| + |01\rangle_{ab}\langle 01| + |10\rangle_{ab}\langle 10| - |11\rangle_{ab}\langle 11| , \tag{5.3}$$

which corresponds to an Ising-type interaction, $\Lambda Z^{(ab)} = e^{-i\pi H^{(ab)}}$, with interaction Hamiltonian $H^{(ab)}$ given by

$$H^{(ab)} = \frac{1}{2}\left(\mathbb{1} - \sigma_z^{(a)}\right) \otimes \frac{1}{2}\left(\mathbb{1} - \sigma_z^{(b)}\right) = |11\rangle_{ab}\langle 11| .$$

That is, $|G\rangle$ is generated from a pure product state by applying interactions between all pairs of particles connected by edges.

We list some useful relations for later reference: The 2^N common eigenstates of the operators $K^{(a)}$ can be generated from $|G\rangle$ by applying all possible products of $\sigma_z^{(a)}, a \in 1, 2, \ldots, N$. This can be seen from

$$\begin{aligned} K^{(b)}\sigma_z^{(a)} |G\rangle &= \underbrace{\sigma_x^{(b)} \prod_{c\in N_b} \sigma_z^{(c)}}_{K^{(b)}} \sigma_z^{(a)} |G\rangle = \sigma_x^{(b)}\sigma_z^{(a)} \prod_{c\in N_b} \sigma_z^{(c)} |G\rangle \\ &= (-1)^{\delta_{ab}}\sigma_z^{(a)} \underbrace{\sigma_x^{(b)} \prod_{c\in N_b} \sigma_z^{(c)}}_{K^{(b)}} |G\rangle = (-1)^{\delta_{ab}}\sigma_z^{(a)} |G\rangle , \end{aligned} \tag{5.4}$$

which means that $\sigma_z^{(a)} |G, \mathbf{0}\rangle = |G, 0\ldots 0\overset{\overset{a}{\downarrow}}{1}0\ldots 0\rangle$. From this relation, together with the fact that $\sigma_y^{(a)} = i\sigma_x^{(a)}\sigma_z^{(a)}$, one can deduce the effect of $\sigma_x^{(a)}$ and $\sigma_y^{(a)}$ (see Ref. [HEB04] and [ADB05] for proofs): Splitting the index vector $\boldsymbol{\mu}$ into μ_a, $\boldsymbol{\mu}_{N_a}$ (neighborhood of vertex a), and $\boldsymbol{\mu}_{R_a}$ (remaining vertices), we write

$$\sigma_z^{(a)} |G, \mu_a \boldsymbol{\mu}_{N_a} \boldsymbol{\mu}_{R_a}\rangle = |G, \overline{\mu_a} \boldsymbol{\mu}_{N_a} \boldsymbol{\mu}_{R_a}\rangle \tag{5.5}$$

$$\sigma_x^{(a)} |G, \mu_a \boldsymbol{\mu}_{N_a} \boldsymbol{\mu}_{R_a}\rangle = (-1)^{\mu_a} |G, \mu_a \overline{\boldsymbol{\mu}_{N_a}} \boldsymbol{\mu}_{R_a}\rangle \tag{5.6}$$

$$\sigma_y^{(a)} |G, \mu_a \boldsymbol{\mu}_{N_a} \boldsymbol{\mu}_{R_a}\rangle = i(-1)^{\overline{\mu_a}} |G, \overline{\mu_a}\ \overline{\boldsymbol{\mu}_{N_a}} \boldsymbol{\mu}_{R_a}\rangle \tag{5.7}$$

where the over-bar means bit complementation.

Graph states in the valence bond solid (VBS) picture

Another description of graph states was introduced in Ref. [VC04b]. In this picture, every edge is replaced by a pair in a maximally entangled state, usually $(|00\rangle + |01\rangle + |10\rangle - |11\rangle)/2$. Each qubit a gets replaced by d_a virtual qubits, where $d_a = |N_a|$ is the degree of vertex a. The physical qubit is recovered by projecting the virtual qubits onto the two-dimensional subspace of the physical one (see Fig. 5.1) using as projector

$$P_d = |\tilde{0}\rangle\langle 0\ldots 0| + |\tilde{1}\rangle\langle 1\ldots 1| . \tag{5.8}$$

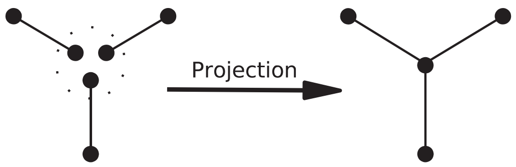


Figure 5.1: Producing a graph state in the VBS picture.

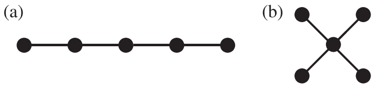


Figure 5.2: Graphs for (a) cluster and (b) GHZ states.

Cluster states and GHZ states

In this article, we study the purification of graph states using two important representatives from this class as examples. By "cluster states", we mean graph states associated with a regular lattice as graph, in this article always a line as in Fig. 5.2a, and with $\boldsymbol{\mu} = \mathbf{0}$. The term GHZ state will *in this article* be used for a graph state (again with $\boldsymbol{\mu} = \mathbf{0}$) associated with a star-shaped graph G_* as in Fig. 5.2b. Such a state can be written as

$$|G_*\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \otimes |+\rangle^{\otimes(N-1)} + |1\rangle \otimes |-\rangle^{\otimes(N-1)}\right)$$

and is hence in its entanglement properties equivalent to an "ordinary" GHZ state $\frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right) = (\mathbb{1} \otimes \mathrm{Had}^{\otimes(N-1)})\,|G_*\rangle$ (where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ and Had is the Hadamard operation $\mathrm{Had} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$).

Bell pairs and graph-state formalism

In order to keep a certain homogeneity, we will employ a new notation for the states of the Bell basis, usually written as:

$$\begin{aligned} |\Phi^\pm\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right) \\ |\Psi^\pm\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right), \end{aligned}$$

Applying a Hadamard operation on the second qubit, one obtains a new basis formed by the graph states $|G_2, 00\rangle$, $|G_2, 01\rangle$, $|G_2, 10\rangle$ and $|G_2, 11\rangle$, where G_2

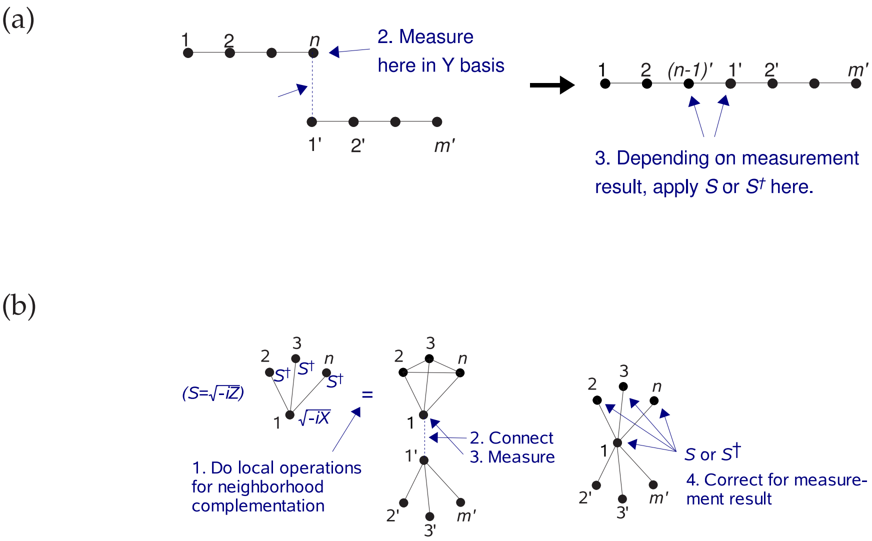

(a)
1
2
n
2. Measure here in Y basis
1'
2'
m'
1
2
(n-1)'
1'
2'
m'
3. Depending on measurement result, apply S or S† here.
(b)
(S=√-iZ)
2
3
n
S†
S†
S†
1
√-iX
=
1. Do local operations for neighborhood complementation
2
3
n
1
1'
2. Connect
3. Measure
2'
3'
m'
2
3
n
1
S or S†
4. Correct for measurement result
2'
3'
m'

Figure 5.3: How to use the connection procedure of Sec. 5.2.1 to assemble large (a) cluster and (b) GHZ states from smaller cluster or GHZ states.

denotes the graph composed of two vertices and one edge. Our new notation shows directly the relation between this two bases:

$$
\begin{aligned}
|\Phi^+\rangle &=: |B;00\rangle &&= \mathrm{Had}^{(2)}\,|G_2,00\rangle\\
|\Psi^+\rangle &=: |B;01\rangle &&= \mathrm{Had}^{(2)}\,|G_2,01\rangle\\
|\Phi^-\rangle &=: |B;10\rangle &&= \mathrm{Had}^{(2)}\,|G_2,10\rangle\\
|\Psi^-\rangle &=: |B;11\rangle &&= \mathrm{Had}^{(2)}\,|G_2,11\rangle\,. \qquad (5.9)
\end{aligned}
$$

Connection of graph-states

In this section, we define a procedure to connect two graph states, $|G_1\rangle$ with N_1 qubits, and $|G_2\rangle$ with N_2 qubits, "fusing" together their respective vertices a_1 and a_2, yielding a state $|G\rangle$ with N_1+N_2-1 qubits. This process is depicted in Fig. 5.3. To realize this action, one applies a projective measurement on a_1 and a_2, given by $P_2 = |0\rangle\langle 00| + |1\rangle\langle 11|$ and $P_2^\perp = |0\rangle\langle 01| + |1\rangle\langle 10|$ (with outcomes 0 and 1). P_2 is defined like in the VBS picture. By similarity with this picture, if the result of the measurement is 0, the final state is the graph state resulting from the connection of G_1 and G_2. If one obtains 1, a correction has to be done. As shown below, it is sufficient to apply $\prod_{b\in N_{a_2}} \sigma_z^{(b)}$ to the resulting state. Recalling that $K_G^{(a)}|G\rangle = |G\rangle$ with $K_G^{(a)} = \sigma_x^{(a)} \prod_{b\in N_a} \sigma_z^{(b)}$, one sees that any graph state can be decomposed as $|G\rangle = |0\rangle_a \otimes |\chi\rangle + |1\rangle_a \otimes \prod_{b\in N_a} \sigma_z^{(b)} |\chi\rangle$. Applying $\prod_{b\in N_{a_2}} \sigma_z^{(b)}$ to the state resulting from $P_2^{\perp}$ one obtains

$$\prod_{b\in N_{a_2}} \sigma_z^{(b)} P_2^{\perp} |G_1\rangle |G_2\rangle$$

$$= \prod_{b\in N_{a_2}} \sigma_z^{(b)} \left(\sigma_z^{(d)} |0\rangle \otimes |\chi_1\rangle \otimes \prod_{d\in N_{a_2}} |\chi_2\rangle + |1\rangle \otimes \prod_{c\in N_{a_1}} \sigma_z^{(c)} |\chi_1\rangle |\chi_2\rangle \right) =$$

$$|0\rangle \otimes |\chi_1\rangle \otimes |\chi_2\rangle + \prod_{b\in(N_{a_1}+N_{a_2})} \sigma_z^{(b)} |1\rangle \otimes |\chi_1\rangle \otimes |\chi_2\rangle = |G_1 + G_2\rangle \quad (5.10)$$

Fig. 5.3 shows how to use this technique to assemble cluster and GHZ states from $|G_2\rangle$ states.

5.2.2 Noise model

Channel noise

In any realistic setting, channels will be noisy. We study the influence of channel noise by considering restricted noise models, where the Kraus representation of the superoperators is diagonal in the Pauli basis. This is a common and usually sufficiently general model [DHCB05] (In particular, any noisy channel can be brought to such a form by means of (probabilistic) local operations). This allows for an efficient and convenient simulation by Monte Carlo techniques (see Sec. 5.4.1). We consider the following channels:

phase-flip channel:

$$\rho \mapsto \mathcal{E}_z^{(a)}(\rho) = q\rho + (1-q)\sigma_z^{(a)}\rho\sigma_z^{(a)} \quad (5.11)$$

bit-flip channel:

$$\rho \mapsto \mathcal{E}_x^{(a)}(\rho) = q\rho + (1-q)\sigma_x^{(a)}\rho\sigma_x^{(a)} \quad (5.12)$$

depolarizing channel:

$$\rho \mapsto \mathcal{E}^{(a)}(\rho) = q\rho + \frac{1-q}{3}\Big(\sigma_x^{(a)}\rho\sigma_x^{(a)} + \sigma_y^{(a)}\rho\sigma_y^{(a)} + \sigma_z^{(a)}\rho\sigma_z^{(a)}\Big) \quad (5.13)$$

In case of depolarizing channel, we define

$$p = (4q - 1)/3$$

which allow us to rewrite Eq. (5.13) as

$$\rho \mapsto \mathcal{E}^{(a)}(\rho) = p\rho + \frac{1-p}{4}\Big(\rho + \sigma_x^{(a)}\rho\sigma_x^{(a)} + \sigma_y^{(a)}\rho\sigma_y^{(a)} + \sigma_z^{(a)}\rho\sigma_z^{(a)}\Big) \quad (5.14)$$

$(1-q)$ will be called the *alteration probability* and p the *reliability*.

Local noise

As part of the purification protocols, local one- and two-qubit unitary operations are employed which may be noisy. An imperfect operation is modeled by preceding the perfect operation $U^{(ab)}$ with the application of one of the noise superoperators $\mathcal{E}$ from Eqs. (5.11–5.13), i. e. the state is transformed as

$$\rho \mapsto U^{(ab)} \left(\mathcal{E}^a (\mathcal{E}^b (\rho)) \right) U^{\dagger (ab)}.$$

We assume that the protocols are executed with the least possible number of operations to keep accumulated noise low. Hence, if a two-qubit gate $U_{12}^{(ab)}$ is preceded by one-qubit gates $U_1^{(a)}$ and $U_2^{(b)}$ we apply one combined unitary $U^{(ab)} = U_1^{(a)} U_2^{(b)} U_{12}^{(ab)}$ which is subjected to noise only once.

Commutation between connection and noise

We now state an observation that will later (in Sec. 5.5.2) be of use.

For any graph states $|G_1\rangle$ and $|G_2\rangle$ which are connected by the procedure described in Sec. 5.2.1, one can show that the noise processes commute with the connection procedure, if they are expressed by a superoperator by only σ_z Pauli operators. This comes from the fact that the neighborhood of the connected vertices a_1 and a_2 changes with the connection and hence, σ_x and σ_y Pauli operators will affect different vertices (see Eqs. (5.5), (5.6) and (5.7)).

The commutation rules between projector P_2 (see Eq. (5.8)) and σ_z can be deduced from the following expression of the connected graph state:

$$\begin{aligned} P_2 |G_1\rangle |G_2\rangle \;\; = \;\; P_2 \;\; \Big(& |0\rangle_{a_1} |0\rangle_{a_2} |\chi\rangle_1 |\chi\rangle_2 + \\ & + \prod_{c \in N_{a_2}} \sigma_z^{(c)} |0\rangle_{a_1} |1\rangle_{a_2} |\chi\rangle_1 |\chi\rangle_2 + \\ & + \prod_{b \in N_{a_1}} \sigma_z^{(b)} |1\rangle_{a_1} |0\rangle_{a_2} |\chi\rangle_1 |\chi\rangle_2 + \\ & + \prod_{b \in N_{a_1}} \sigma_z^{(b)} \prod_{c \in N_{a_2}} \sigma_z^{(c)} |1\rangle_{a_1} |1\rangle_{a_2} |\chi\rangle_1 |\chi\rangle_2 \Big) \end{aligned}$$

Recalling that $\sigma_z |0\rangle = |0\rangle$, $\sigma_z |1\rangle = - |1\rangle$ one can show that:

$$P_2 \sigma_z^{(a_1)} |G_1\rangle |G_2\rangle \;\; = \;\; \sigma_z^{(a)} P_2 |G_1\rangle |G_2\rangle \qquad (5.15)$$

$$P_2 \sigma_z^{(a_2)} |G_1\rangle |G_2\rangle \;\; = \;\; \sigma_z^{(a)} P_2 |G_1\rangle |G_2\rangle \qquad (5.16)$$

5.2.3 Local noise equivalent

To judge how close state ρ it to the desired state $|\psi\rangle$, one often usually employs the fidelity $F := \langle\psi| \rho |\psi\rangle$. However, is may be advantageous to regauge the fidelity by introducing the following derived measure: We define the local noise equivalent (LNE) as the level of local depolarizing noise (in terms of the alteration probability $(1-q)$ of Eq. (5.13)) that one has to apply to each qubit of the perfect state $|\psi\rangle$ to deteriorate it to the same fidelity F as ρ has. The advantage of this measure is twofold: (i) It is more natural for uses of states in quantum error correction schemes, as it can be compared directly to the fault-tolerance threshold in case of uncorrelated-noise models. (ii) It does not fall off exponentially with the size of a state for constant noise levels, as the fidelity does. On the other hand, it often cannot be calculated analytically in a straight-forward way. We hence used a numerical Monte Carlo simulation of the state deterioration (which is why the LNE scale in the figures has error bars).

5.2.4 Purification protocols

The purpose of entanglement purification is the following: One is given an ensemble of multi-party states, which all are distributed over two (or more) sites and exhibit entanglement between the sites. These states are only an approximation to the desired state $|\Psi\rangle\langle\Psi|$ with an insufficient fidelity, which one wishes to improve. As the sites are spatially separated, one cannot apply joint operation on the distributed parts of a state. Instead, one compares (in case of the so-called recurrence protocols, which are considered here solely) pairs of entangled states, makes joint operations on them, and then measures one of the state in order to gain information about the other. Only for specific measurement outcomes, the other state is kept. After iterating this procedure, one is left with an ensemble of smaller number of particles but higher fidelity.

Bipartite purification

Several protocols have been proposed to purify bipartite entangled states [BBP+96, BDSW96, DEJ+96]. To test the different strategies, we used the most efficient which can be used to purify an ensemble of $|\Phi^+\rangle$ states, namely the one described in Ref. [DEJ+96]. We present here a modified version of this bipartite entanglement purification protocol (BEPP) which allows for the purification of the connected graph-state pair. As we are concerned only with this graph in this section, we simply write $|\mu\nu\rangle$ for the different basis states $|G_2,\mu\nu\rangle$. Recall that $|00\rangle = 1/\sqrt{2}\,(|0\rangle\,|+\rangle + |1\rangle\,|-\rangle)$ (see Eq. (5.9)).

Alice and Bob want to share entangled pairs with high fidelity. At the beginning they are given an ensemble of noisy $|0,0\rangle$ states, each of them owning one part of the pairs. We consider a state diagonal in the graph-state basis,

$$\rho = x_{00}|00\rangle\langle 00| + x_{01}|01\rangle\langle 01| + x_{10}|10\rangle\langle 10| + x_{11}|11\rangle\langle 11|. \tag{5.17}$$

We remark that such a standard form can always be achieved by means of depolarization, i.e. applying certain (random) local unitary operations. Each step of the protocol consists of the following operations: (i) Alice and Bob perform unitary operations on their particles, with Alice's and Bob's unitaries given by

$$S_A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}, \quad S_B = \frac{1}{\sqrt{2}} \mathrm{Had} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \mathrm{Had}^\dagger .$$

(ii) Alice performs a CNOT operation from the first state to the second and Bob from the second state to the first; (iii) Alice and Bob measure the second state in different bases. To see the effect of this procedure, we calculate the fidelity and yield obtained after one step with two initial states given by (5.17).

In (i), Alice and Bob apply S_A and S_B, respectively, in order to swap $|11\rangle$ and $|10\rangle$. Then, in step (ii), they apply the bilateral CNOT. One can check that the effect of this operation on the graph state basis is given by the following map:

$$|\mu_A \mu_B\rangle \, |\nu_A \nu_B\rangle \mapsto |\mu_A \oplus \nu_A, \mu_B\rangle \, |\nu_A, \nu_B \oplus \mu_B\rangle \,, \tag{5.18}$$

(Here, $\oplus$ indicates bitwise AND, i. e. addition modulo 2.) Last (iii), Alice and Bob measure the qubits of the target state. This is done in the eigenbasis $\{|0\rangle_x, |1\rangle_x\}$ of σ_x for Alice and in the computational basis $\{|0\rangle_z, |1\rangle_z\}$, for Bob. By this they obtain the eigenvalue of the correlation operator K_2 defined in Eq. (5.1) and determine the value of the second bit describing the state. If it is 0, they keep the first state. They discard it otherwise.

After the measurement, they keep the control state with success probability $k = (x_{00} + x_{11})^2 + (x_{01} + x_{10})^2$ and the new coefficients are given by:

$$\begin{aligned} x'_{00} &= (x_{00}^2 + x_{11}^2)/k \\ x'_{01} &= (x_{01}^2 + x_{10}^2)/k \\ x'_{10} &= (2x_{00}x_{11})/k \\ x'_{11} &= (2x_{01}x_{10})/k. \end{aligned} \tag{5.19}$$

Hence, the fidelity is $F = x'_{00} = (x_{00}^2 + x_{11}^2)/k$. The yield of the step, defined as the number of remaining states divided by the number of states before the step, is given by $k/2$ as half of the states (the targets) are measured and discarded.

The unitary operations performed at the beginning of the protocol (step (i)) are required for its convergence. It guarantees that fidelity 1 is a fix point of the protocol which is approached when iterating the procedure. The CNOT operation is a means of transferring information from the first qubit to the second. The measurement allows to distinguish between $\{|0,0\rangle, |1,0\rangle\}$ and $\{|0,1\rangle, |1,1\rangle\}$ and hence, determines the second bit of the index vector.

Multipartite purification

Multipartite purification protocols (MEPP) have been introduced in Ref. [MPP+98] for GHZ states, were further developed in Ref. [MS02] and extended to all two-colourable graph states in Refs. [DAB03, ADB05]. Recall that a two-colourable graph-state is a graph state in which the vertices can be separated into two sets V_A and V_B such that no edges exist between vertices of the same set. Using the procedure described in Ref. [ADB05], one can depolarize any mixed state ρ to the form

$$\rho = \sum_{\boldsymbol{\mu}_A,\boldsymbol{\mu}_B} \lambda_{\boldsymbol{\mu}_A,\boldsymbol{\mu}_B} |G,\boldsymbol{\mu}_A,\boldsymbol{\mu}_B\rangle\langle G,\boldsymbol{\mu}_A,\boldsymbol{\mu}_B| \tag{5.20}$$

without changing the diagonal coefficients (where $\boldsymbol{\mu}_A$, $\boldsymbol{\mu}_B$ are binary vectors corresponding to sets V_A, V_B respectively). Hence we will restrict our attention to input states of this form. The protocol is composed of two subprotocols P1 and P2 which we will describe here:

Subprotocol P1: The states composing the ensemble of two-colourable graph-states are processed pair-wise. All parties belonging to set V_A perform a CNOT operation from the second state of a pair of states to the first one while the parties belonging to set V_B perform a CNOT from the first one to the second one. This leads to the transformation

$$|G,\boldsymbol{\mu}_A,\boldsymbol{\mu}_B\rangle\,|G,\boldsymbol{\nu}_A,\boldsymbol{\nu}_B\rangle \mapsto |G,\boldsymbol{\mu}_A,\boldsymbol{\mu}_B\oplus\boldsymbol{\nu}_B\rangle\,|G,\boldsymbol{\nu}_A\oplus\boldsymbol{\mu}_A,\boldsymbol{\nu}_B\rangle\,, \tag{5.21}$$

As in the bipartite protocol, the last step consists of measuring the second state of the pair. The parties belonging to set V_A measure their qubit a in the eigenbasis $\{|0\rangle_x, |1\rangle_x\}$ of σ_x, obtaining results $\xi_a \in \{0,1\}$, while the ones belonging to set V_B make their measurement in the computational basis, obtaining results $\zeta_b \in \{0,1\}$. From this, we can calculate the part of the index vector of the measured state (second state of the r. h. s. of Eq. (5.21)) corresponding to set V_A:

$$\boldsymbol{\nu}'_A = \boldsymbol{\nu}_A \oplus \boldsymbol{\mu}_A = \left(\xi_a \oplus \bigoplus_{b\in N_a} \zeta_b\right)_{a\in V_a}.$$

If this is $\mathbf{0}$, it is most likely that $\boldsymbol{\mu}_A = \mathbf{0}$, and hence, the first state is kept (and otherwise discarded). As consequence, in the expansion (5.20) of the ensemble density matrix, elements of the form $\lambda_{\mathbf{0},\boldsymbol{\mu}_B}$ are increased. One finds that the new matrix elements are given by

$$\lambda'_{\boldsymbol{\gamma}_A,\boldsymbol{\gamma}_B} = \frac{1}{\kappa} \sum_{\{(\boldsymbol{\nu}_B,\boldsymbol{\mu}_B)|\boldsymbol{\nu}_B\oplus\boldsymbol{\mu}_B=\boldsymbol{\gamma}_B\}} \lambda_{\boldsymbol{\gamma}_A,\boldsymbol{\nu}_B}\lambda_{\boldsymbol{\gamma}_A,\boldsymbol{\mu}_B} \tag{5.22}$$

where κ is a normalization constant such that $\mathrm{tr}(\tilde{\rho}) = 1$.

Subprotocol P2: As explained above, subprotocol P1 is employed to purify with respect to the eigenvalues $\boldsymbol{\mu}_A$ associated with set V_A. The second subprotocol leads to the purification with respect to the eigenvalues of set V_B. It is obtained from P1 by exchanging the roles of set V_A and V_B. The protocol's action is described by the following map:

$$|G, \boldsymbol{\mu}_A, \boldsymbol{\mu}_B\rangle\,|G, \boldsymbol{\nu}_A, \boldsymbol{\nu}_B\rangle \mapsto |G, \boldsymbol{\mu}_A \oplus \boldsymbol{\nu}_A, \boldsymbol{\mu}_B\rangle\,|G, \boldsymbol{\nu}_A, \boldsymbol{\nu}_B \oplus \boldsymbol{\nu}_B\rangle\,. \tag{5.23}$$

One measures the second state. The measurements on set V_B are done in the eigenbasis $\{|0\rangle_x, |1\rangle_x\}$ of σ_x while they are done in the computational basis in set V_A. This leads to the determination of part $\boldsymbol{\mu}_B$ of the index vector. As in subprotocol P1, one keeps the state if $\boldsymbol{\mu}_B = \mathbf{0}$. The new coefficient are given by

$$\lambda'_{\gamma_A,\gamma_B} = \frac{1}{\kappa} \sum_{\{(\boldsymbol{\nu}_A,\boldsymbol{\mu}_A)|\boldsymbol{\nu}_A \oplus \boldsymbol{\mu}_A = \gamma_A\}} \lambda_{\boldsymbol{\nu}_A,\gamma_B} \lambda_{\boldsymbol{\mu}_A,\gamma_B} \tag{5.24}$$

where κ is a normalization constant such that $\mathrm{tr}(\tilde{\rho}) = 1$, as before.

5.3 Strategies

5.3.1 Quantum communication cost $C_{F,G}$

We now define our figure of merit, the quantum communication cost. We consider N spatially separated parties A_k, $k = 1, 2, \ldots, N$ which are pairwise connected by noisy quantum channels $\mathcal{E}_{kl}$, described by completely positive maps acting on density operators for qubits. We will quantify the quantum communication through these quantum channels using the quantum communication cost C_{kl}, i.e. the number of usages of the quantum channel $\mathcal{E}_{kl}$, while classical communication between pairs of parties will be considered to be for free. Sending a single qubit through the quantum channel $\mathcal{E}_{kl}$ costs 1 unit, i.e. $C_{kl} = 1$, while the transmission of an arbitrary state of M qubits costs $C_{kl} = M$. We will be interested in the total quantum communication cost C, where

$$C = \sum_{k<l} C_{kl}. \tag{5.25}$$

We consider the generation of multipartite entangled states (graph states, to be specific) $|G\rangle$ distributed among the parties A_k. The goal is to generate states $\Lambda = \bigotimes_{i=1}^{L} \rho_i$, where the fidelity of each ρ_i, $F_i = \langle G|\,\rho_i\,|G\rangle$, fulfills $F_i \geq F$. That is, each of the states has a fidelity larger than a threshold value F, which we call the "desired target fidelity". We remark that we demand that the ensemble of output states are in a tensor product form. In principle, weaker requirements such as that only the reduced density operators of Λ have fidelity larger than F are conceivable, however one faces certain difficulties

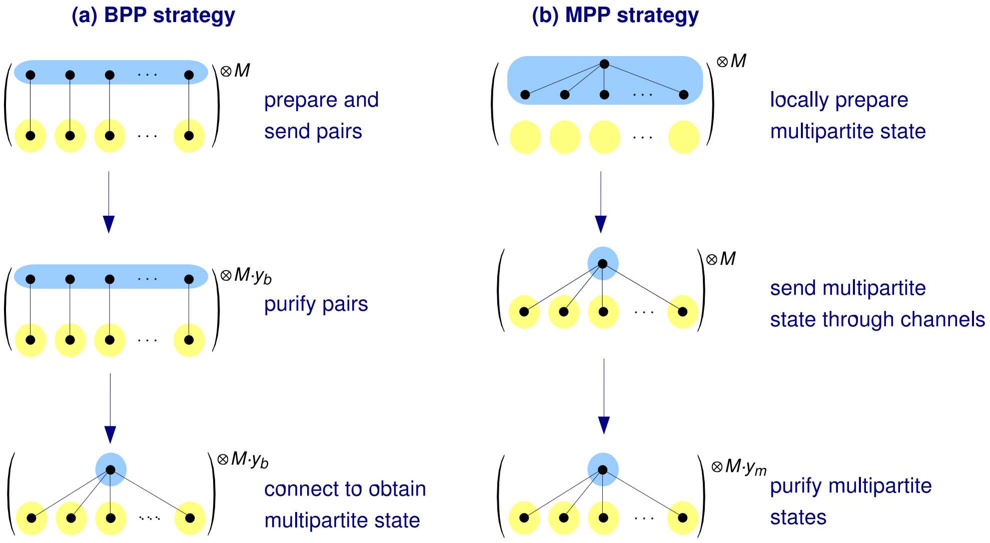

(a) BPP strategy
(b) MPP strategy
⊗M
prepare and send pairs
⊗M
locally prepare multipartite state
⊗M·y_b
purify pairs
⊗M
send multipartite state through channels
⊗M·y_b
connect to obtain multipartite state
⊗M·y_m
purify multipartite states

Figure 5.4: Distribution of N-qubit GHZ states over noisy channels. (a) Bipartite entanglement purification strategy: Bell pairs are sent over the channels and purified using a BEPP. The purified pairs are then connected (using the procedure of Fig. 5.3) to the desired GHZ state. (b) Multipartite entanglement purification strategy: Alice prepares the GHZ state locally and sends all but one of the particles through the channels. Then, the MEPP protocol is used.

in this case. For instance, it is not clear whether each of the copies of the state can be independently used for further quantum information processing tasks due to possible classical correlations among the copies. Hence, we deliberately demand the tensor product structure. We will be interested in the total quantum communication cost C required to generate $\Lambda = \bigotimes_{i=1}^{L} \rho_i$ with $F_i \geq F$. In particular, we consider the quantum communication cost per copy,

$$C_{F,G} = \frac{C}{L}, \tag{5.26}$$

where one optimizes over all possible strategies to generate Λ. Due to this optimization, the quantity $C_{F,G}$ is very difficult to calculate. Hence we restrict ourselves to establish upper bounds on $C_{F,G}$ by considering explicit strategies to generate high fidelity multipartite entangled states.

Multiple variations of this problem are conceivable. For simplicity we will assume that all parties are pairwise connected by identical quantum channels, $\mathcal{E} = \mathcal{E}_{kl}$. Inhomogeneous situations where only some pairs of parties are connected by quantum channels (a restricted communication web), or where the classical communication is limited, or cases where quantum channels between different pairs of parties are different (i.e. different noise parameter) will not be considered here.

We will look mainly at two scenarios depicted in Fig. 5.4, which we describe now.

5.3.2 Bipartite purification strategy

In the BEPP strategy, the parties A_k, $k = 1, 2, \ldots N$ wish to create a shared ensemble of N-qubit graph states of high fidelity using a BEPP, where party A_k holds the qubit corresponding to vertex a_k. For each edge of the graph, one of the two parties connected by this edge prepares a connected graph-state pair, $|G_2, 00\rangle$ (equivalent to a Bell pair up to a local unitary) and sends one qubit of the pair to the other party through a noisy channel. (Alternatively, one could use a teleportation-based strategy: Alice distributes Bell pairs to the $N-1$ other parties. The pairs are purified and then used to distribute the multipartite state that Alice has prepared locally.) The effect of the channels is given by Eq. (5.13) leading to states of fidelity $F = q + \frac{1-q}{4}$ and diagonal in the graph-state basis. The parties repeat the operation M times so that at the end $M|E|$ entangled pairs are distributed between the different partners, where $|E|$ is the number of edges in the graph. The BEPP (reviewed in Section 5.2.4) is then applied. This leads to a smaller ensemble of states given by a density matrix of the same form but with higher fidelity. Finally, the connection procedure described in Sec. 5.2.1 is applied: Each party A_k merges together the $|N_{a_k}|$ qubits which will connect vertex a_k with its neighbors leading to the desired graph state. We call

$$Y(F) = \frac{\#\text{ final states}}{\#\text{ initial states}} = \frac{\#\text{ final states}}{M}$$

the yield of the production of final states with fidelity F. To build up the desired multipartite state $|G\rangle$, we need one $|G_2\rangle$ pair for each edge of G. The number of edges for 1D cluster and GHZ states is (as for any tree graph) $|E| = N-1$. Hence, the quantum communication cost is related to the yield by

$$C_{F,G} = \frac{N-1}{Y(F)}. \tag{5.27}$$

The numerator is the number of channel uses (i. e. number of transmitted qubits) required to distribute one state. This dependence on the size of the state properly reflects that for larger states, already the preparation of the raw states is more costly. To allow for easier comparison with the yield, a figure that may feel more familiar to the reader, we have plotted in all graphs the *inverse* communication cost $C_{F,G}^{-1} = Y(F)/(N-1)$ which is proportional to the yield.

5.3.3 Multipartite purification strategy

Alternatively, in the MEPP strategy, a central party, called Alice, creates M N-qubit graph states locally. For each graph state, she keeps one qubit and sends the other $N-1$ qubits through the channels to the $N-1$ other parties. The resulting states are then purified using direct multipartite entanglement purification, i. e. the MEPP reviewed in Sec. 5.2.4. Hence, to distribute one state, we need $N-1$ channel uses, the same as in the BEPP case. Thus, Eq. (5.27) holds for MEPP, as well.

5.3.4 Mixing of strategies

Assume that the application of, say, m steps of one of the protocols mentioned above reaches a final fidelity F_1 with a communication cost of C_1, and application of $m+1$ steps achieves fidelity $F_2 > F_1$ with communication cost $C_2 > C_1$ (i. e. $Y_2 < Y_1$). For a certain application, a fidelity of F with $F_1 < F < F_2$ is required, i. e. m steps are insufficient, but $m+1$ steps achieve a higher fidelity than desired at the cost of lower yield. In this case, one can find a compromise between the two strategies by mixing ensembles:

Choosing an $\alpha \in [0,1]$, one prepares M raw states and then uses the first strategy on αM of them in order to gain $\alpha M Y_1$ states of fidelity F_1, and the second strategy on the remaining $(1-\alpha)M$ states to obtain $(1-\alpha)MY_2$ states of fidelity F_2. Mixing these states gives an ensemble of fidelity

$$F = \frac{\alpha Y_1 F_1 + (1-\alpha) Y_2 F_2}{\alpha Y_1 + (1-\alpha) Y_2} \tag{5.28}$$

with a a yield $Y = \alpha Y_1 + (1-\alpha) Y_2$. This method allows one to obtain intermediate fidelities with a better yield. The communication cost mixes according to

$$\frac{1}{C} = \frac{\alpha}{C_1} + \frac{1-\alpha}{C_2}. \tag{5.29}$$

5.3.5 Intermediate strategies

As a “compromise” between BEPP and MEPP, we shall also consider the following set of strategies: Assemble small states of N_1 qubits, send them through the channels, purify them, and then use the connection scheme described in Sec. 5.2.1 to connect L of the N_1-qubit states to one state with $N = LN_1 - L + 1$ qubits.

5.4 Numerical simulations

5.4.1 Technique

In generic cases, an explicit numerical simulation of a quantum systems is intractable due to the exponential growth of the Hilbert space with the number of involved particles or qubits. In our case, however, an efficient simulation is possible for two reasons: (i) All gates that are employed by the protocols are elements of the so-called Clifford group and hence, the Gottesman-Knill theorem applies, which allows for efficient simulations of pure state evolutions. (ii) The considered noise channels have Kraus representations that are diagonal in the Pauli basis.

To explain (i), we start by reviewing the Gottesman-Knill theorem [Got98a, NC00]. It says that it is possible to simulate so-called stabilizer circuits efficiently on a classical computer. These are quantum circuits containing only preparation of computational basis states, operations from the Clifford group, and measurements in the computational basis. The N-qubit Clifford group $\mathcal{C}_N$ is the group of those unitary operations that map Pauli operators onto Pauli operators under conjugation, i. e.

$$\mathcal{C}_N := \left\{ U \in SU(2^N) \mid UPU^\dagger \in \mathcal{P}_N \quad \forall P \in \mathcal{P}_N \right\}, \\ \mathcal{P}_N := \{\pm 1, \pm i\} \cdot \{\mathbb{1}, \sigma_x, \sigma_y, \sigma_z\}^{\otimes N}. \quad (5.30)$$

It happens to contain all the operations that we need for purifying, and hence, we can simulate the execution of the purification protocols described in Section 5.2.4.

Aaronson and Gottesman have given a fast algorithm which can perform such a simulation, and also supplied an implementation in the C programming language [AG04]. We have used this software at the beginning of our studies, but after realizing that its performance is not sufficient for our purposes, developed a new, faster algorithm, which is described elsewhere [AB06].

The state represented in our simulator is always a pure state.[2] However, in entanglement purification, one usually deals with mixed states, represented as density matrices. Nevertheless, due to the fulfillment of condition (ii), we can get around this problem using a Monte Carlo technique, which we describe now.

To represent the ensembles of states we start with a high number N_i of qubits, typically several thousand times the number of qubits in the states to be purified. The qubits are initialized to a tensor product of $|G, \mathbf{0}\rangle$ states. Note that all these qubits can potentially get entangled, and hence have to be part of the same simulated quantum register. This would be prohibitive without a very efficient algorithms for the stabilizer simulation.

We then simulate all steps that are required to prepare Bell pairs or graph states, to purify them and to measure them. Depending on the measurement results, states are kept or discarded. Several iterations of the protocols are simulated.

The transmission through the perfect channels amount to a simple relabeling: The program remembers the new site, where the qubit resides, as this indicates which qubits can be subject of joint operations.

---

[2] There is an algorithm for simulation the evolution of a rather restricted class of mixed states [AG04], which is however not general enough for our purposes.

Simulating the channel noise is done by randomizing over many simulation runs as follows. The three noisy channels that we have considered, Eqs. (5.11-5.13), are simulated using a pseudo-random number generator (RNG). Whenever noise is to be applied onto a qubit, a random number between 0 and 1 is generated, and if it is smaller than $(1-q)$ (the noise level), σ_x (σ_z) is applied for bit-flip (phase-flip) noise. For depolarizing noise, the RNG is used again to obtain an integer between 1 and 4 which determines which of the operators $\mathbb{1}, \sigma_x, \sigma_y, \sigma_z$ to apply.

After the preparation of M initial states, m iterations of the protocol and for the BEPP case, connection of the purified pairs, N_f final states remain. The yield is then given by $\tilde{Y} = \frac{N_f}{M}$. This is, however, not a good estimate for the asymptotic yield in the limit of infinite ensembles for the following reason: If the number N_{i-1} of states at the beginning of purification step i is odd, we have to discard one state, because we can only deal with pairs of states. Hence, we better estimate the yield by

$$Y = \frac{N_1}{\llbracket M \rrbracket} \cdot \frac{N_2}{\llbracket N_1 \rrbracket} \cdot \ldots \cdot \frac{N_f}{\llbracket N_{m-1} \rrbracket} \tag{5.31}$$

(with $M \equiv N_0$, $N_f \equiv N_m$, and $\llbracket N \rrbracket = N$ for even N, and $\llbracket N \rrbracket = N-1$ for odd N.)

The fidelity can be determined by measuring the final states in the graph basis. This is because all the intended operations and the random noise operations map graph states onto graph states, so that all N_f final states are of the form $|G, \boldsymbol{\mu}\rangle$. The index $\boldsymbol{\mu}$ can be determined as follows: For each state, the graph state creation operation of Eq. (5.2), $\prod_{\{a,b\}\in E} \Lambda Z^{(ab)}$ (which is Hermitean) is applied again onto the state. If one then applies Hadamard gates on all qubits and measures in the σ_z basis, the measurement results spell out the index vector $\boldsymbol{\mu}$. As we intended to create $|G, \mathbf{0}\rangle$ states, we call the number of states for which we measured $\mathbf{0}$ the number N_g of "good" states and hence estimate the fidelity as

$$F = \frac{N_g^{\mathrm{tot}}}{N_f^{\mathrm{tot}}} \pm \sqrt{\frac{N_g^{\mathrm{tot}}(N_f^{\mathrm{tot}} - N_g^{\mathrm{tot}})}{(N_f^{\mathrm{tot}})^3}}.$$

The superscript "tot" indicates that many runs of the simulations are made and that the numbers are the sums of the numbers in the individual runs. The uncertainty term follows from the expectation that, given a true fidelity F_T, the number of good states F_g^{tot} output by the Monte Carlo simulations after many runs is distributed according to a binomial distribution with length N_f^{tot}, hit probability F_T and hence standard deviation $N_f^{\mathrm{tot}}\sqrt{F_T(1-F_T)}$. Thus, $\frac{N_g}{N_f}$ is the estimate for F_T with the given statistical uncertainty at 1σ level.

In the same way, the yield can be assigned an uncertainty,[3]

$$Y = \frac{N_f^{\text{tot}}}{M^{\text{tot}}} \pm \sqrt{\frac{N_f^{\text{tot}}(M^{\text{tot}} - N_f^{\text{tot}})}{(M^{\text{tot}})^3}}.$$

The 1σ uncertainties are indicated by error bars in the plots.

5.4.2 Extremal strategies

We now present the results obtained for the two extremal strategies described in Secs. 5.3.2 and 5.3.3 with the following parameters: The distribution of the qubits is done through noisy channels and each step of protocol requires imperfect two-qubit operations. The noise considered is depolarizing noise as defined in Eq. (5.13) with reliability $p = 0.9$ (10% noise) for the channels and $p_l = 0.99$ (1% noise) for the local operations. We used the Monte-Carlo simulation method described in Sec. 5.4.1 to reach a precision on the fidelity varying from 1‰ to 1% depending on the size of the states and the number of iterations.

For the MEPP case, one has to decide, which sequence of the sub-protocols P1 and P2 to use. The alternating sequence, P1-P2-P1-. . . , turns out to be not optimal in terms of yield and fidelity, neither for GHZ nor for cluster states. To find the optimum, one might hence consider to simulate, after each step, both sub-protocols, and then continue with the better one. Somewhat surprisingly, this leads to worse results (see Fig. 5.5). Thus, to find the optimal sequence of m protocol-steps, one would need to try all 2^m possibilities. As this is not practical, we decided to stick with the alternating sequence, which turned out, though it is not optimal, to give very descent performance. For GHZ states, there is also a difference between the alternating sequences P1-P2-P1-. . . and P2-P1-P2-. . . , due to the asymmetry of the sets V_A (containing only Alice's qubit) and V_B (containing the rest). Starting with P1 works better, and this is what we use in all plots discussed in this section.

GHZ states

We start with the results obtained for GHZ states. We made our simulations for states of three to 10 qubits and a maximum number of steps varying from 5 to 7. As an example, Fig. 5.6 shows the quantum communication cost as a function of desired fidelity for 5-qubit GHZ states. The data points are the outputs for 1 to 6 steps of the protocol. This plots allow us to determine, for a given fidelity, the strategy which will give the best yield (lowest communication cost).

---

[3] To be precise, we should calculate the uncertainty not from $Y = \frac{N_f}{M}$, but following Eq. (5.31). This simplification is however justified, as it only increases the uncertainty estimate, and this only slightly.

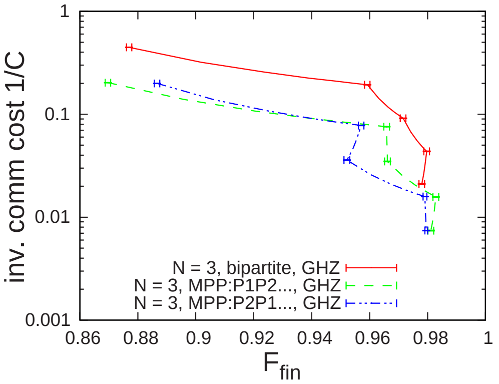


Figure 5.5: Comparison of the values of inverse of communication cost and fidelity obtained after a number of steps varying from 1 to 5 for 3-qubit GHZ states. The red solid line stands for the BEPP strategy, the green dashed line for the alternating sequence of MEPP subprotocols beginning with P1 and the blue small dashed line for the alternating sequence beginning with P2.

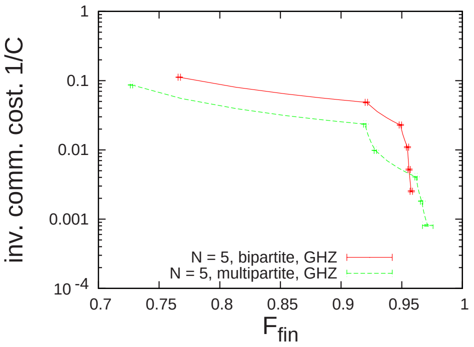


Figure 5.6: Inverse of communication cost for different target fidelities for 5-qubit GHZ states and with $p = .9$ and $p_l = .99$ (where p is the reliability defined in Eq. (5.14)). The data points are the outputs for $1, 2, \ldots, 6$ iterations of the protocol. The connecting lines are obtained by mixing ensembles of different fidelities according to Eq. (5.29). The red solid line gives the obtained value in the MEPP and the green dashed line for the BEPP. The gain on fidelity from one step to the other becomes smaller at each step. From 6 to 7 steps, the gain in fidelity is smaller than the uncertainty both in the BEPP and in the MEPP strategy. We consider this value as the maximal reachable fidelity.

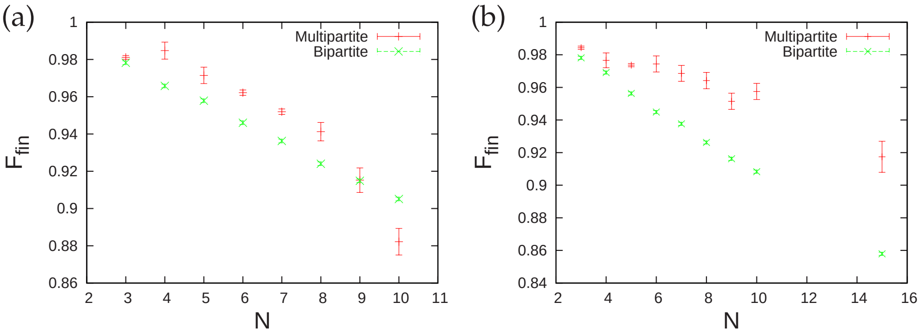


Figure 5.7: Maximal reachable fidelity as function of N for (a) GHZ and (b) cluster states for the bipartite (green $\times$) and multipartite (red bars) strategies with reliabilities (cf. Eq. (5.14)) $p = 0.9$ for channel transmission and $p_l = 0.99$ for local operations. The final fidelity is estimated as follows: For a given number of parties, we iterated the protocol as long as we obtained an increase of fidelity larger than the uncertainty (typically 1%). We took the last value as maximal fidelity and assigned its uncertainty to the maximal reachable fidelity. The green crosses give the values in the bipartite case while the red bars give it for the multipartite case. One sees here the main difference in behavior between GHZ and cluster states. In the first case, there is a range where the multipartite strategy is better than the bipartite one for a number of parties strictly smaller than 10. For more parties, the multipartite protocol fails because of the fragility of GHZ states against noise. On the other hand, the robustness of cluster states allow us to purify them even for a large number of parties. The range of fidelity where MEPP is superior increases with the number of parties.

After 6 iterations, the increase in fidelity obtained by an additional step is smaller than the chosen precision of 1%. We therefore take this value as estimate of the maximum reachable fidelity. A comparison of the maximal reachable fidelity for both strategies as a function of the number of parties (Fig. 5.7a) shows that the maximal reachable fidelity is higher in the MEPP case for a number of parties strictly smaller than 10. In this case, there is a transition value of target fidelity from which on the MEPP strategy gives a better yield. We will refer to the value pair of fidelity and communication cost, where this transition happens, as the cross-over point. Fig. 5.8 presents the yield as function of fidelity for $N = 3$ and $N = 10$ as well as the cross-over points for intermediate number of parties. $N = 9$ is the highest number of qubits for which there is a cross-over point. For higher number of parties, the BEPP strategy is always better. This is because of the fragility against noise of GHZ states for large particle numbers [DB04, HDB05]. The communication cost and fidelity of the cross-over points as function of the number of parties are presented in Fig. 5.9a.

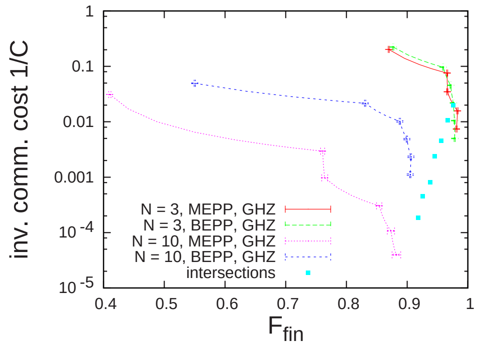


Figure 5.8: Inverse of communication cost for different target fidelities of GHZ states of 3 to 10 qubits with alteration probabilities (as in Fig. 5.7) $1 - p = 0.1$ and $1 - p_l = 0.01$. The dashed green line stands for 3-qubit GHZ states and BEPP strategy, the red solid line for 3-qubit GHZ states and MEPP strategy, the blue small-dashed line for 10-qubit GHZ states and BEPP strategy, the pink dotted line for 10-qubit GHZ states and MEPP strategy. The blue squares give the cross-over points, i.e. the fidelity where MEPP becomes more efficient than BEPP, for $N = 3, 5, 6, 7, 8$ and 9. For $N = 3$ and $N = 10$, the purification curves are plotted as well. For $N = 3$, they cross at the corresponding blue square.

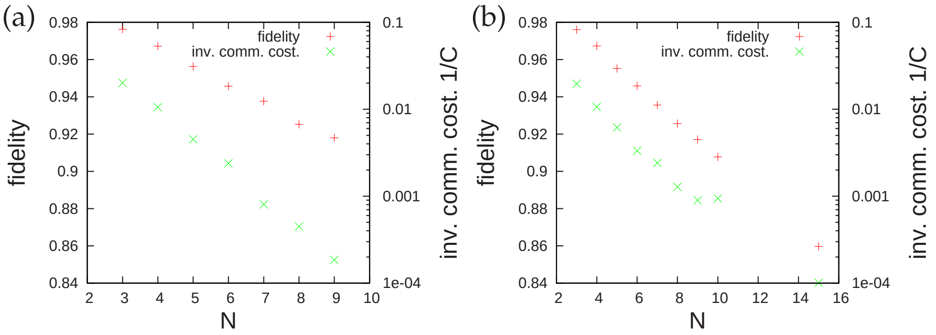


Figure 5.9: Inverse of communication cost (green ×) and fidelity (red +) of the cross-over depending on the number of parties for (a) GHZ and (b) cluster states. This values are obtained by using the Monte Carlo method and are therefore submitted to errors. The cross-over indicates the range of target fidelity from which up the MEPP strategy is more efficient than the BEPP strategy. Note the log scale for the inverse of communication cost. In the GHZ case, there is no cross-over for more than 9 parties.

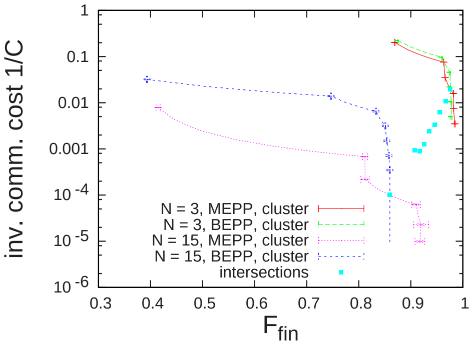


Figure 5.10: Inverse of communication cost for different target fidelities for 3 (red solid line for MEPP and green dashed line for BEPP) and 15 (pink dotted line for MEPP and blue small dashed line for BEPP) qubit cluster states. The data points are the outputs for 1, 2, 3, . . . iterations of the protocol. The intermediate points are obtained by mixing ensembles of different fidelities. For more than 6 steps, the difference between the reached fidelity and the maximum reachable fidelity is smaller than the uncertainty. For any number of parties, the curves representing the two strategies cross over. The disks give this cross-over for $N = 3, 4, 5, 6, 7, 8, 9, 10, 15$. (That one curve seems to "go back" is just an artifact of the statistical inaccuracies of the Monte Carlo method.)

Cluster states

Next, we did simulations cluster states using the same parameter as for the GHZ states. The results are quite different.

We made our simulations for states of three to fifteen qubits. In this range, as one can see in Fig. 5.10, there is always a cross-over point. This is in stark contrast with the GHZ case. This main difference in behavior between this two kind of states is due to the much higher robustness of cluster states against noise [HDB05, HDB05]. Moreover, the range of target fidelity for which the multipartite strategy is the only one available increases with the number of parties as shown in Fig. 5.7b. In Fig. 5.9b, we present the fidelity and communication cost of the cross-over point. Both values decrease with the number of parties. This is due to the increasing cost of producing bigger and bigger states and also to the fact that we consider here the global fidelity and not the LNE presented in Sec. 5.2.3.

5.4.3 Intermediate strategies

Since switching from BEPP to MEPP can result in such striking differences in yield, one might expect that, especially near the break-even point, certain

Mn	prepare an n-qubit cluster or GHZ state
B2	prepare a Bell pair
S	send the states through the channels
P1	apply multipartite purification protocol P1
P2	apply multipartite purification protocol P2
Pb	apply bipartite protocol
Cℓ	connect ℓ states to a larger one

Table 5.1: Legend for the instruction strings in Figs. 5.11 and 5.13.

intermediate strategies, mixing characteristics of BEPP and MEPP, might perform even better. After all, in the BEPP case, one purifies small states (with only 2 qubits) and then connects them, while in the MEPP scenario, the states are first connected to large units, which are then purified. One can also connect pairs to states of intermediate size, purify these, connect them to the desired full size, and perhaps purify again. This can be seen e. g. in Fig. 5.11. In this figure, we have simulated many different strategies which are described in short by instruction strings which are processed from left to right and tell the software in which order which preparations, transmissions, connections or purifications should be simulated (cf. Table 5.1).

It can be seen that for low fidelities and high yields (left side of the plots), the BEPP case is best, as already seen above, and for high fidelities and low yields (right margin of the plots), MEPP catches up. In the middle region, one may indeed increase the performance by first preparing small states of, say, 4, 5 or 7 qubits, purifying them, and then connecting them to the desired 13-qubit state. (Do not get confused by the appearance of "M13-S" at the left margin. This looks like MEPP, but is not, as it contains no purification at all. Also note that there is a subtle difference between using the BEPP protocol (denoted "B2-S-Pb-. . . ") and using the MEPP protocol on the $|G_2\rangle$ state (denoted "M2-S-P1-. . . " or "M2-S-P2-. . . "), with the former performing better.)

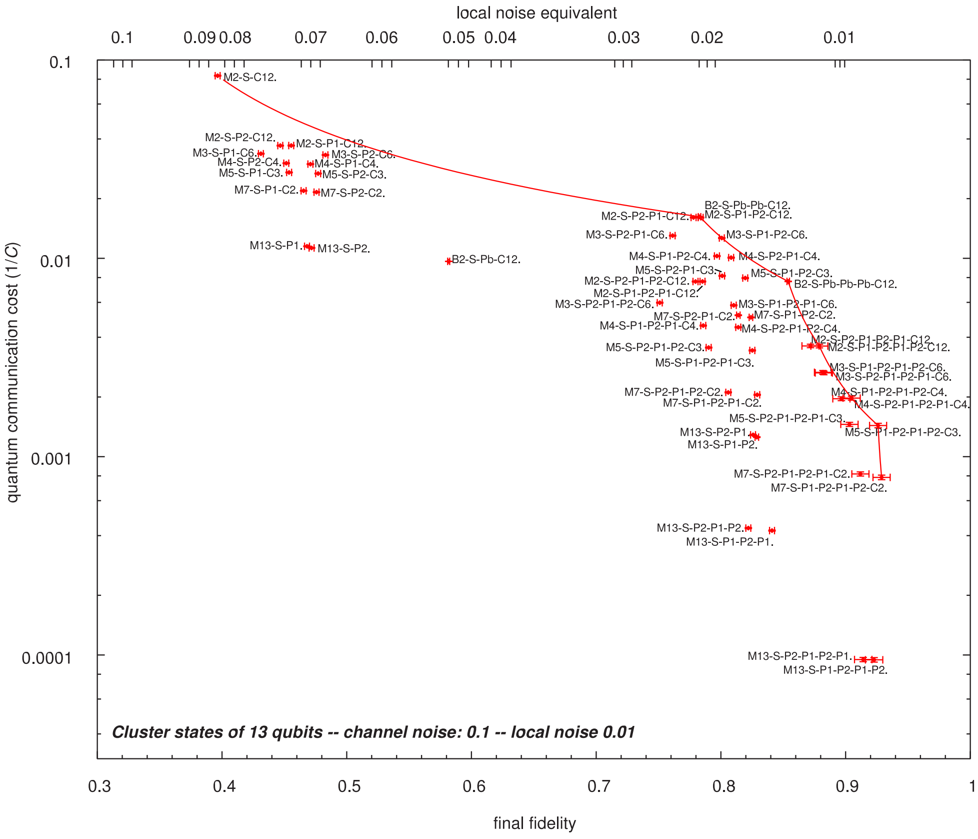
Of course, only discrete ways of assembling the desired states from equal smaller states are available. Recall that connecting L states of n qubits will give a state of

$$N = Ln - (L-1) \tag{5.32}$$

qubits because $(L-1)$ qubits have to be measured in the connection process. In the plots, we have taken all possible values of L for the given state size n and calculated data points for the corresponding strategies with up to four purification steps. The blue curve in the plot marks the optimum that can be achieved using theses strategies, and mixing them as described in Section 5.3.4.

To demonstrate the efficiency of our procedure, we also considered pu-

(a)



(b)

See page 102 for subfigure (b).

Figure 5.11: Examples for the use of intermediate purification strategies, here for 13-qubit (a) cluster and (b) GHZ states. Plotted is the inverse of communication cost as function of final fidelity. See Table 5.1 for the meaning of the instruction strings. The blue curve marks the maximal achievable yield for a given desired fidelity and is obtained by connecting the optimal strategies with curves according to Eq. (5.28). Noise levels are $(1-q) = 0.1$ for the channels and $(1-q_l) = 0.01$ for local operations. In (a), one can –following the dark blue line– see well, how for small target fidelity . . . *[cont. on page 102]*

(a)

See page 101 for subfigure (a).

(b)

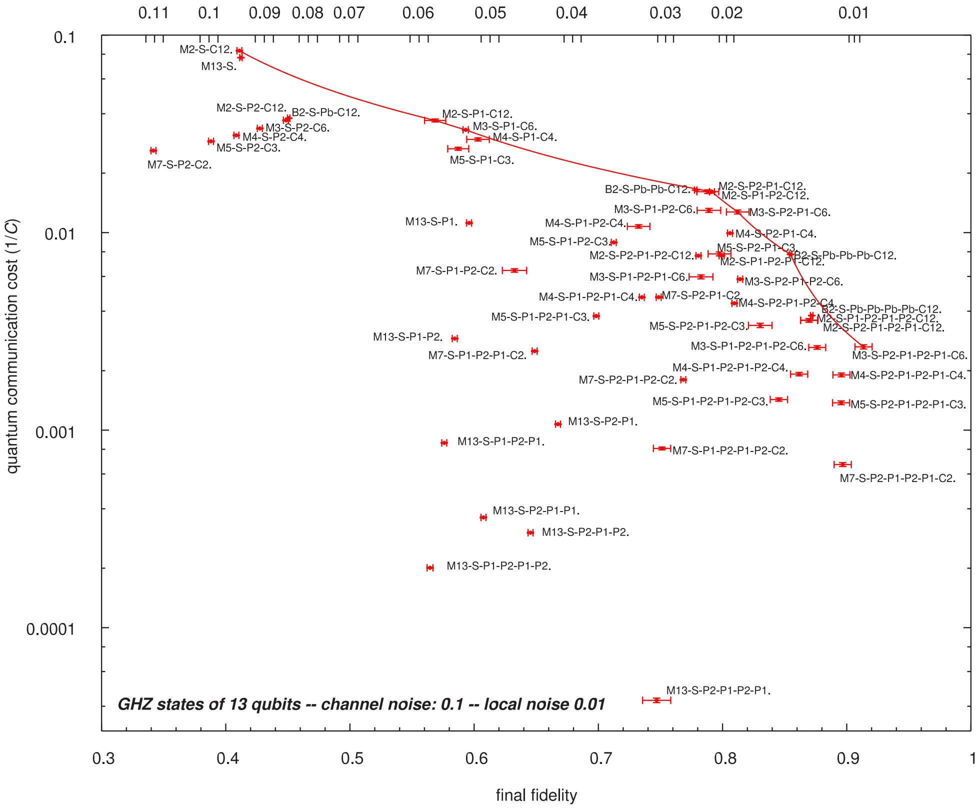


Figure 5.12: *[cont. from page 101]* . . . BEPP ("M2-S-. . . ") gives the best yield, while for high fidelities ($F \gtrsim 0.9$), distributing larger and larger states becomes advantageous. (For even higher fidelities, one expects the full MEPP strategy, i. e. "M13-S-. . . ", to appear on the blue curve. However, this will happen at communication costs larger than the scales shown on the plot, which ends with "M13-S-P1-P2-P1", i. e. MEPP with only three purification steps.) In (b), the picture is not as clear, at the GHZ states already start to deteriorate under the given level of local noise.

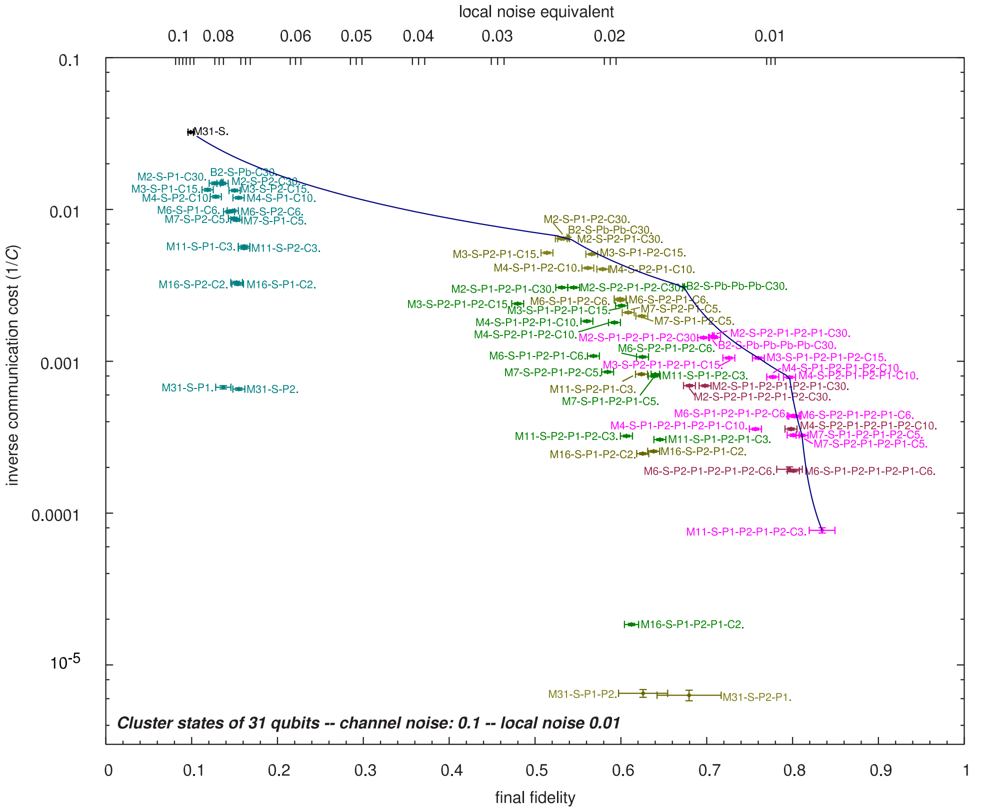


Figure 5.13: Production of 31-qubit cluster states, using intermediate strategies. The "instruction strings" are explained in Table 5.1. Data points with the same number of purification steps are plotted in the same color. Note how the distribution of initially larger states becomes advantageous for higher target fidelities. Noise levels are $(1-q) = 0.1$ for the channels and $(1-q_l) = 0.01$ for local operations. (Note also that the the data points at the low end of the plot have few purification steps: Those steps beginning with "M16" or "M31" that should appear on the curve of optimal strategies are again, as in Fig. 5.11, beyond the range of the plot.)

rification of cluster states of 31 qubits (Fig. 5.13). In order to allow for easier comparison, Fig. 5.13, as well as Fig. 5.11, show a re-gauging of the fidelity axis to the so-called local noise equivalent (LNE) described in Sec. 5.2.3.

5.5 Analytical treatment for a simplified model

For a better understanding of the numerical results, we now develop an analytical treatment for both BEPP and MEPP. To make this task feasible we have to restrict ourselves to a simplified noise model. We only consider GHZ states.

As before, we define two sets V_A and V_B corresponding to the bi-colouration of the graph. V_A is the set containing only one qubit, namely the central vertex which is connected to all the others, and V_B contains the rest.

In the toy models presented below, the central party, called Alice wants to share an N-qubit GHZ state with $(N-1)$ partners. Depending on the strategy, the initial states are either Bell pairs or GHZ states, which are noisy due to the transmission through the channels. First, in Subsection 5.5.1, we consider local operations to be perfect. We will see that this fails to reproduce features seen in the numerical results. Hence, we extend our model, in Section 5.5.2, such that it incorporates local noise.

5.5.1 Perfect local operations

To start, we assume to local operations to be perfect. Of the channels we considered in Eqs. (5.11-5.13), only bit-flip channels and phase-flip channels allow for a simple analytical treatment. We present the calculation for phase-flip channels. The calculation and the results for bit-flip channels are very similar. We have hence not included them in the paper.

BEPP strategy

Following the BEPP scenario described in Sec. 5.3.2, Alice sends one qubit of each entangled pair $\rho = |G_2; 0,0\rangle\langle G_2; 0,0|$of her initial ensemble through the channel to party B_k, obtaining

$$\rho = q|G_2; 0,0\rangle\langle G_2; 0,0| + (1-q)|G_2; 0,1\rangle\langle G_2; 0,1|.$$

She then applies the BEPP. The state of the pairs that are kept after one step is given by (see Eq. 5.19)

$$\rho = \frac{q^2}{q^2 + (1-q)^2}|G_2; 0,0\rangle\langle G_2; 0,0| + \frac{(1-q)^2}{q^2 + (1-q)^2}|G_2; 0,1\rangle\langle G_2; 0,1|. \quad (5.33)$$

In this step, the probability of keeping the source state after the measurement of the other state is given by $k^{\text{BEPP}}(q,1) = q^2 + (1-q)^2$ (probability of having same measurement outcomes). We denote by $f^{\text{BEPP}}(q,m)$ the fidelity after m steps. The quantity $k(q,m)$ is called success probability in step m. Note that the ratios of the ensemble sizes after and before the step is given by $k(q,m)/2$ as one half of the states are measured and discarded. One obtains the total yield Y^{BEPP} after m steps by multiplying these ratios for the individual steps. By iterating the protocol over m steps one finds:

$$\begin{aligned} f^{\text{BEPP}}(q,m) &= \frac{q^{2^m}}{q^{2^m} + (1-q)^{2^m}}, && (5.34) \\ k^{\text{BEPP}}(q,m) &= \frac{q^{2^m} + (1-q)^{2^m}}{\left[q^{2^{m-1}} + (1-q)^{2^{m-1}}\right]^2}, && (5.35) \\ Y^{\text{BEPP}}(q,m) &= \prod_{i=1}^{m} \frac{k(q,i)}{2} \\ &= \frac{q^{2^m} + (1-q)^{2^m}}{2^m \prod_{i=1}^{m-1} \left(q^{2^i} + (1-q)^{2^i}\right)}. && (5.36) \end{aligned}$$

After the bipartite purification, Alice connects $(N-1)$ pairs to produce an N-qubit GHZ state. To connect two pairs, she applies a controlled phase gate (Eq. (5.3)) followed by a σ_y measurement on one of the two qubits just connected (cf. Fig. 5.3). This procedure is repeated $(N-1)$ times between different pairs of parties (A, B_k), $k = 1, \ldots, N-1$, in order to obtain the N-qubit GHZ state.

Note that the qubits that Alice connects have not been sent through channels and are hence unaffected by channel noise. Thus, it does not matter whether we first apply the superoperator for the channel noise and then the one for the local noise due to the connection process, or *vice versa*. This means that the final state is obtained by applying noise on all qubits of the GHZ state that do not belong to Alice. This leads to a fidelity

$$F^{\text{BEPP}}(N,q,m) = f^{\text{BEPP}}(q,m)^{N-1}$$

and (as the channels are used $N-1$ times to create one N-qubit GHZ state) a quantum communication cost

$$C^{\text{BEPP}} = \frac{N-1}{Y^{\text{BEPP}}(q,m)}.$$

MEPP strategy

In the MEPP setting, Alice prepares an N-qubit GHZ state locally and distributes it through depolarizing channels to her $(N-1)$ partners. We then have the state

$$\rho^{(0)} = \left(\prod_{a=2}^{N} \mathcal{E}_z^{(a)} \right) |G_*; 0, \mathbf{0}\rangle\langle G_*; 0, \mathbf{0}|, \tag{5.37}$$

where $\mathcal{E}_z^{(a)}$ is the phase-flip channel defined in Eq. (5.11). We shall from now on suppress the symbol G_* which indicates the N-vertices star graph of Fig. 5.2.

We shall see that all states that we encounter have the form

$$\begin{aligned}\rho^{(m)} = r_0^{(m)} |\mathbf{0}\rangle\langle\mathbf{0}| + & \\ + r_1^{(m)} \sum_{i_1=2}^{N} |0, \ldots 0\overset{\overset{i_1}{\downarrow}}{1}0 \ldots 0\rangle\langle 0, \ldots 0\overset{\overset{i_1}{\downarrow}}{1}0 \ldots 0| + & \\ + r_2^{(m)} \sum_{\substack{i_1, i_2 = 2 \\ i_1 < i_2}}^{N} |0, \ldots 0\overset{\overset{i_1}{\downarrow}}{1}0 \ldots 0\overset{\overset{i_2}{\downarrow}}{1}0 \ldots\rangle\langle 0, \ldots 0\overset{\overset{i_1}{\downarrow}}{1}0 \ldots 0\overset{\overset{i_2}{\downarrow}}{1}0 \ldots| + & \\ + \cdots + r_{N-1}^{(m)} |0, 11 \ldots 1\rangle\langle 0, 11 \ldots 1|, &\end{aligned} \tag{5.38}$$

where $r_j^{(m)}$ denotes the coefficient in front of the terms with j entries "1" after the mth step of the purification protocol. These states are diagonal in the graph state basis and symmetric w. r. t. permutations of the qubits in set V_B. They are hence characterized by only N coefficients $r_0^{(m)}, \ldots, r_{N-1}^{(m)}$.

We start by carrying out the application of the superoperator in Eq. (5.37). Indeed, one obtains a mixture of the form (5.38) with coefficients

$$r_j^{(0)} = q^{N-1-j} (1-q)^j. \tag{5.39}$$

As only set V_B is affected by the noise, subprotocol P2 is sufficient to purify the state. Following [ADB05], one sees that after each step of the subprotocol, the state is changed such that each coefficient becomes proportional to the square of its former value, i. e.

$$r_j^{(m)} = \frac{\left[r_j^{(m-1)}\right]^2}{\sum_{i=0}^{N-1} \binom{N-1}{i} \left[r_i^{(m-1)}\right]^2}. \tag{5.40}$$

Inserting Eq. (5.39), one gets for the first step (using the binomial theorem)

$$r_j^{(1)} = \frac{q^{2(N-1-j)}(1-q)^{2j}}{[q^2 + (1-q)^2]^{N-1}},$$

and iterating the formula, one finds

$$r_j^{(m)} = \frac{q^{2^m(N-1-j)}(1-q)^{2^m j}}{[q^{2^m} + (1-q)^{2^m}]^{N-1}}.$$

The fidelity of the state at step m can now be read off:

$$F^{\text{MEPP}}(N,q,m) = r_0^{(m)} = \left[\frac{q^{2^m}}{q^{2^m} + (1-q)^{2^m}}\right]^{N-1} .$$

Note that this is the same expression as we got before for the BEPP: $F^{\text{MEPP}}(N,q,m) = F^{\text{BEPP}}(N,q,m)$.

To calculate the yield, we need the success probability $k^{\text{MEPP}}(N,q,m)$ that a state is kept. Using a similar argument as we did for Eq. (5.40) we find

$$k^{\text{MEPP}}(N,q,m) = \sum_{i=0}^{N-1} \binom{N-1}{i} \left[r_i^{(m-1)}\right]^2 = \left(\frac{q^{2^m} + (1-q)^{2^m}}{\left[q^{2^{m-1}} + (1-q)^{2^{m-1}}\right]^2}\right)^{N-1} .$$

From this, we can find the yield as before

$$\begin{aligned} Y^{\text{MEPP}}(N,q,m) &= \prod_{i=1}^{m} \frac{k^{\text{MEPP}}(N,q,i)}{2} \\ &= 2^{-m} \left(\frac{q^{2^m} + (1-q)^{2^m}}{\prod_{i=1}^{m-1} \left[q^{2^i} + (1-q)^{2^i}\right]}\right)^{N-1} \end{aligned} \tag{5.41}$$

Comparing to the BEPP case, one sees that

$$Y^{\text{MEPP}}(N,q,m) = 2^{m(N-2)} \left[Y^{\text{BEPP}}(N,q,m)\right]^{N-1} . \tag{5.42}$$

Conclusion

In the particular case of dephasing channels and perfect local operations, both strategies lead to the same fidelity after iterating the protocol the same number of steps. However, they differ in the communication cost. As one sees from Eq. (5.42), the yield of the MEPP strategy is always lower (and the communication cost hence larger). This fact can be explained from the higher probability of throwing away states at each step, which even increases further with the number of parties.

We have also done analytical calculations for bit-flip channels (Eq. (5.11)) and numerical simulations for depolarizing channels (Eq. (5.13)) (always in case of perfect local operations) and found a similar behavior.

In order to see regions where MEPP is superior as we did with the Monte Carlo simulations, it is hence necessary to give up the simplification of assuming noiseless local operations.

5.5.2 Imperfect local operations

If local operations are not assumed to be perfect, results are quite different. We again consider GHZ states of arbitrary size, to be purified with the BEPP or MEPP strategy.

For the noise, we define a model which is simple enough to allow for analytical calculations but still shows the general features obtained numerically (especially it shows cross-over points): The channel through which Alice sends qubit to the other parties is the phase-flip channel of Eq. (5.11) with alteration probability $(1-q)$. The imperfection of the local gates are modeled by bit-flip noise (Eq. (5.12)), for Alice's operations, and phase-flip noise for all other parties, always with alteration probability $(1-q_l)$.

MEPP strategy

As before, Alice prepares perfect N-qubit GHZ state and distributes them using the channels. We get the same initial state $\rho^{(0)}$ as before, again of the form (5.38) with coefficients as in Eq. (5.39). We shall see that, again, the form (5.38) is preserved by the purification steps even though they are now assumed to be noisy.

The values of the r_i are changed according to a linear map:

$$r_j^{(m)} = \sum_{k=0}^{N-1} \Lambda_{jk} r_k^{(m-1)}.$$

We shall construct this map in two steps. First, we see, how the phase-flip noise of the local gates acting on the qubits in V_B,

$$\prod_{i=2}^{N} \mathcal{E}_z^{(i)}(q_l), \tag{5.43}$$

changes the coefficients and denote the map corresponding to this action by λ:

$$r'_j = \sum_{k=0}^{N-1} \lambda_{jk} r_k^{(m-1)}.$$

Then, we consider the action of the bit-flip noise on Alice's qubit get the full map Λ.

For the first step, we call a state $|G_*, \boldsymbol{\mu}\rangle$ a k-state, if $\boldsymbol{\mu}$ starts with a 0 (for the central qubit in V_A) and contains, within the indices corresponding to V_B, k entries "1" and $(N-1-k)$ entries "0". We can now calculate the probability $p_{j\leftarrow k}$ that the superoperator (5.43) changes a pure k-state to any j-state: Say, s of the k entries "1" are flipped to "0". Then $\bar{s} = j-k+s$ of the $(N-1-k)$ entries "0" have to be flipped to "1". Hence,

$$p_{j\leftarrow k} = \sum_{s=0}^{k} \binom{k}{s} (1-q_l)^s q_l^{k-s} \times \binom{N-1-k}{\bar{s}} (1-q_l)^{\bar{s}} q_l^{N-1-k-\bar{s}}. \tag{5.44}$$

There are $\binom{N-1}{j}$ j-states and $\binom{N-1}{k}$ k-states, and so

$$\lambda_{jk} = \frac{\binom{N-1}{k}}{\binom{N-1}{j}} p_{j\leftarrow k},$$

which can also be written in terms of Gauß's hypergeometric function F as

$$\lambda_{jk} = \frac{j!}{k!(j+k)!}\, F\left(\frac{(1-q_l)^2}{q_l^2}; -k, j-N+1, j+k+1\right) \times \\ \times (1-q_l)^{j-k} q_l^{N-1-j+k} \quad (5.45)$$

Now, we can do the second step and apply the noise for the imperfection of Alice's local gates, modeled by $\mathcal{E}_x^{(1)}(q_l)$. We get, due to Eq. (5.6),

$$r_j^{(m)} = q_l r'_j + (1-q_l) r'_{\tilde{N}-j} = \sum_{k=0}^{N-1} \Lambda_{jk} r_k,$$

where

$$\Lambda_{jk} = \frac{\binom{N-1}{k}}{\binom{N-1}{j}} \sum_{s=0}^{k} \binom{k}{s} \left[\binom{N-1-k}{j-k+s} q_l^{N-1-j+k-2s+1} (1-q_l)^{j-k+2s} + \right. \\ \left. + \binom{N-1-k}{j-s} q_l^{N-1-j-k+2s} (1-q_l)^{j+k-2s+1} \right].$$

The fidelity and the yield corresponding to one step of protocol can then be calculated:

$$f(N,q,q_l,m) = \frac{\left[r_0^{(m)}\right]^2}{\sum_{k=0}^{N-1} \binom{N-1}{k} \left[r_k^{(m)}\right]^2} = \frac{\left(\sum_{j=0}^{N-1} \Lambda_{0j} r_j^{(m-1)}\right)^2}{\sum_{k=0}^{N-1} \binom{N-1}{k} \left(\sum_{j=0}^{N-1} \Lambda_{kj} r_j^{(m-1)}\right)^2}.$$

As before, the denominator of the previous expression is the success probability $k(N,q,q_l,m)$ of the step m,[4] and we get the total yield $Y^{\mathrm{MEPP}}(N,q,q_l,m)$ by multiplying up the factors $k/2$ of all m steps:

$$Y^{\mathrm{MEPP}}(N,q,q_l,m) = \prod_{i=1}^{m} \frac{k(N,q,q_l,i)}{2}$$

(For the corresponding quantum communication cost, Eq. (5.27) has to be used.)

BEPP strategy

Next, we find an analytical treatment for the BEPP strategy. In order to facilitate the calculation, we will consider the BEPP as a special case of the MEPP. We first show why this is possible without changing the results:

Considering the restricted noise model presented in this Section, the state can always be written as a contribution of $|G_2;0,0\rangle\langle G_2;0,0|$ and

---

[4] Note that Λ is dependent on m although we have suppressed this to keep notation simple.

$|G_2;0,1\rangle\langle G_2;0,1|$ only. It can hence be purified using only subprotocol P2. In addition, the only difference between the BEPP protocol as described in Sec. 5.2.4 and subprotocol P2 is the exchange of the states $|G_2;1,0\rangle$ and $|G_2;1,1\rangle$ in the former, and not in the latter. As these states have no contribution in the present model, the two protocols give identical results.

The results obtained in the last section can be used to calculate the fidelity $f(q,q_l,m)$ and the yield $Y^{\mathrm{BEPP}}(q,q_l,m)$ before connection. After purification, Alice connects qubits $a_1,\ldots,a_{N-1}$ of $(N-1)$ pairs described by the states $\rho_k = q\,|G_2\rangle_k\langle G_2| + (1-q)\sigma_z^{(b_k)}\,|G_2\rangle_k\langle G_2|\,\sigma_z^{(b_k)}$ respectively. The joint state of the pairs is given by

$$\begin{aligned}\rho =& q^{N-1}\left(\bigotimes_{k=1}^{N-1}|G_2\rangle_k\langle G_2|\right)\\ &+(1-q)\,q^{N-2}\sum_{i=1}^{N-1}\sigma_z^{(b_i)}\left(\bigotimes_{k=1}^{N-1}|G_2\rangle_k\langle G_2|\right)\sigma_z^{(b_i)}+\ldots\\ &+(1-q)^{N-1}\left(\prod_{i=1}^{N-1}\sigma_z^{(b_i)}\right)\left(\bigotimes_{k=1}^{N-1}|G_2\rangle_k\langle G_2|\right)\left(\prod_{i=1}^{N-1}\sigma_z^{(b_i)}\right).\end{aligned}$$

The connection is performed using the procedure described in Sec. 5.2.1. As the noise contains only σ_z operators, it commutes with the projectors and correction operators that are used in this procedure. The state after projection is given by

$$\begin{aligned}P_2\,\rho =& q^{N-1}|G_*\rangle\langle G_*|\\ &+(1-q)\,q^{N-2}\sum_{b=2}^{N}\sigma_z^{(b)}|G_*\rangle\langle G_*|\sigma_z^{(b)}+\cdots+\\ &+(1-q)^{N-1}\left(\prod_{b=2}^{N}\sigma_z^{(b)}\right)|G_*\rangle\langle G_*|\left(\prod_{b=2}^{N}\sigma_z^{(b)}\right).\end{aligned}$$

It follows that after the connection, the fidelity is given by

$$F^{\mathrm{BEPP}} = f(q,q_l,m)^{N-1}$$

and the quantum communication cost by

$$C^{\mathrm{BEPP}} = \frac{N-1}{Y^{\mathrm{MEPP}}(2,q,q_l,m)}.$$

Discussion

The results obtained the way just explained are presented in Figs. 5.14, 5.15, and 5.16, always for MEPP and BEPP.

The cross-over points, i. e. the fidelity values (and corresponding costs) above which MEPP performs better than BEPP, are plotted as blue squares in Fig. 5.14 for up to 70 parties. We have also plotted the fidelity-cost function for selected numbers N of parties, which cross in their respective blue squares. The other cross points were determined in the same way by observing where the MEPP and the BEPP curve intersect. Compare this figure with Fig. 5.8: Our analytical toy model could reproduce the appearance of cross points which we had already discussed in Section 5.4.2, an essential feature observed for the more general noise model. It does not, however, reproduce the fact that a cross-over ceases to appear above a certain number (here: 9) of parties. This fact is due to the particular kind of noise of our toy model under which GHZ states appear less fragile than under depolarizing noise so that the break-down of MEPP for large states does not happen.

We can use our analytic model to explore the parameter space more thoroughly. For instance, one might be interested how the positions of the cross points change if the local noise is increased. This is shown in Fig. 5.15 where the right-most curve is the same as the disks in Fig. 5.14 and the others are for higher local noise levels. Observe how the effect of local noise depends more and more on the state size as its level approaches the order of magnitude of the channel noise.

The vertical tails of the curves in Fig. 5.14 already allow to easily read off the maximum reachable fidelity, which is plotted in Fig. 5.16. There, the advantage of MEPP over BEPP increases with the number of parties. This effect can also be seen in the numerical calculations for depolarizing noise (Fig. 5.7a). In the latter case, it is, however, soon overcome by the competing effect of the break-down of MEPP under realistic (depolarizing) noise.

5.5.3 Testing the numerics

The analytical formulas are also very useful for verifying the code of our numerical calculations. Switching the programs from depolarizing noise to the simplified noise considered here is a trivial alteration. We find that the numerical results agree well with the analytics, see Fig. 5.17 to 5.18. This fact makes us confident in the correctness of our codes.

5.6 Summary and conclusions

In this article, we have investigated the quantum communication cost of preparing a class of multipartite entangled states with high fidelity. The presence of noisy quantum channels and imperfect local control operations requires the usage of error correction or –in our case– entanglement purification schemes to achieve this aim. We have considered various strategies to generate these high–fidelity states and have established in this way upper bounds on the quantum communication cost. The optimal strategy strongly

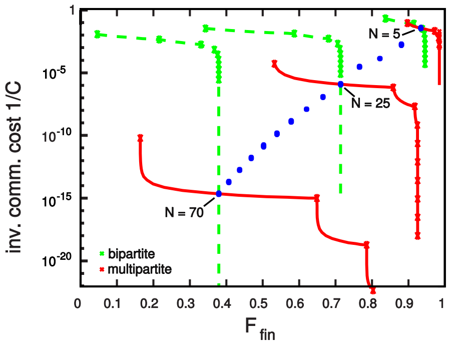


Figure 5.14: Inverse of communication cost as function of final fidelity for the simplified noise model used in section 5.5.2. Analytical calculation for GHZ states of different number of qubits N varying from 5 to 70, with alteration probability for the channel and local noise of $(1-q) = 0.1$ and $(1-q_l) = 0.05$ respectively. The green dashed lines stand for MEPP strategy while the red solid lines stand for BEPP strategy. The blue circles give the crossing points for all number of parties between 5 and 70.

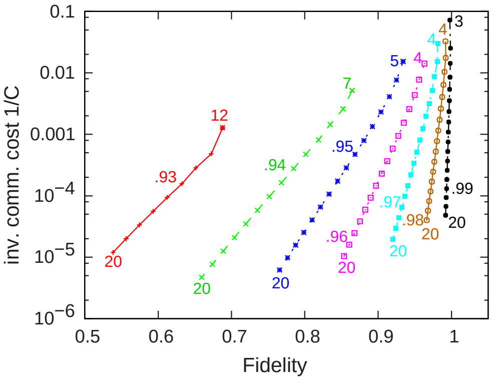


Figure 5.15: Analytical results for different number of parties and different amount of local noise. Each curve gives the yield as function of fidelity for the cross-over for a given alteration probability $(1-q_l)$ (See Eq. (5.11) and (5.12)). This parameter varies from $q_l = 0.93$ (left curve) to $q_l = 0.99$ (right curve).

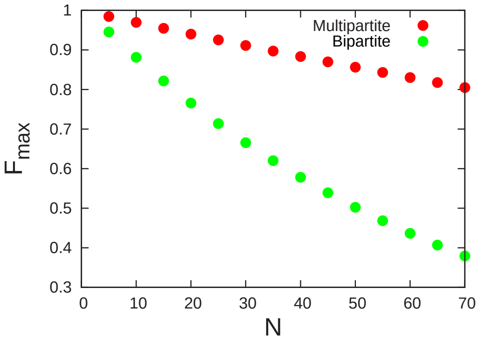


Figure 5.16: Maximal reachable fidelity F_{max} plotted against the number of parties for the simplified model described in Sec. 5.5.2 applied to GHZ states. The alteration probability for the channels and the local noise are given by $(1-q) = 0.1$ and $(1-q_l) = 0.01$ respectively. The results were obtained analytically.

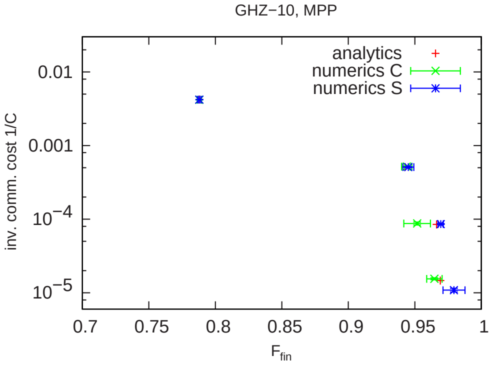


Figure 5.17: Testing the numerics: We switched the programs that were used to calculate the results of section 5.4.2 (program C) and section 5.4.3 (program S) to the simplified noise model of section 5.5.2. The plot shows the inverse of communication cost as function of the final fidelity. The red symbols (+) stand for the analytical results while the green (×) and the blue ($\ast$) symbols stand for the output of program C and program S respectively. The error bars stand for 1σ errors. A comparison with the derived analytical formulas shows satisfactory agreement. The calculation was done for MEPP of GHZ-states with 10 qubits (b) at noise levels $q = 0.9$ and $q_l = 0.95$.

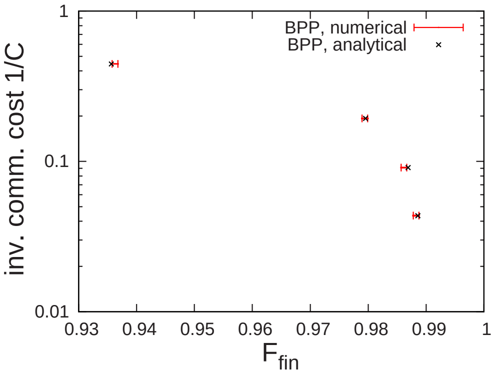


Figure 5.18: Testing the numerics obtained for the BEPP strategy. Noisy entangled pairs, arising from sending one of the qubits through a depolarizing channel, are purified using the BEPP. The plot shows the inverse of communication cost as function of final fidelity. The black crosses give the exact values while the red bars give the numerical results of the Monte-Carlo simulation.

depends on the error parameters for channels and local control operations, and on the desired target fidelity. For a simple error model and the generation of GHZ states based on various strategies, we have obtained analytic results that allow us to compare these strategies. Numerical simulations for generic error models, based on Monte Carlo simulation, show essentially the same features as observed in the simplified model. The simulation makes use of a recently developed method that allows one to efficiently simulate the evolution of stabilizer states (or graph states) under Clifford operations on a classical computer [AB06, AG04]. We have also applied this method to investigate not only the generation of GHZ states but also of other types of multipartite entangled states, e.g. cluster states.

We find that for high target fidelities, strategies based on multipartite entanglement purification generally perform better than strategies based on bipartite purification. For low target fidelities, strategies based on bipartite purification have a higher efficiency, leading to smaller communication cost.

We believe that the generation of high–fidelity multipartite entangled states is of significant importance in the context of (distributed) quantum information processing. Such multipartite entangled states represent resources, e. g. for measurement-based quantum computation, conference key agreement and secret sharing schemes, and may be used for other security tasks. Our investigation takes both channel noise and noisy apparatus into account. We could show that the choice of a proper strategy not only allows one to significantly reduce the quantum communication cost, but to reach fidelity of target state that are not accessible otherwise.

Acknowledgments

This work was supported in part by the Austrian Science Foundation (FWF), the Deutsche Forschungsgemeinschaft (DFG), the European Union (IST-2001-38877, -39227, OLAQUI, SCALA). W. D. is supported by project APART of the ÖAW. The numerical calculations have been carried out on the Linux compute cluster of the University of Innsbruck's Konsortium Hochleistungsrechnen.

Part II

Simulations over continuous sets of states

Chapter 6

Overview on the numerical treatment of strongly-correlated systems

Strongly correlated quantum systems are a major challenge to contemporary theoretical physics. Whenever a perturbative treatment cannot give satisfying results, a number of sophisticated numerical techniques may be tried. In this chapter, I attempt to give a brief overview of the most common and important ones of these. Even though these methods have given rise to impressive successes, many phenomena in solid-state physics still wait for an explanation and many researchers feel that exploration of these systems by numerical means will turn out to be crucial. Often given examples for this are high-T_c superconductors and quantum effects in the magnetical properties of solids. (For an introduction to the research on such systems, see e. g. [Aue94].) An important aim in this area of research is to deepen the understanding of quantum phase transitions, i e., phase transitions that are not driven by thermal but by quantum fluctuation and hence occur even at zero temperature. (See [Voj03] for a review.)

Model systems of quantum spins on a lattice are most suitable for such studies. Given that magnetism is explained by spins within a crystal lattice, these models are a most natural abstraction. Already the simple classical XY model[1] (with its critical special case, the Ising model) can become quite difficult, even more so when made a quantum system by adding a transversal field. Onsager's analytic solution of the 2D Ising model without field [Ons44] is considered an early milestone in the field (that then invalidated, to everybody's surprise, most that was thought to be established about phase transitions) and seemingly still marks the end of what is possible without numerics. One of the most challenging of the models of fundamental interest is surely the Heisenberg antiferromagnet, whose ground-state was quickly

---

[1] See Sec. 8.5.1 for definitions of these models.

found to deviate subtly but crucially from the simple Néel order (i. e., neighboring spins are anti-aligned) that one might naively expect and whose low-energy behaviour defeats any easy explanation with quasi-particles due to the model's criticality. Interest in the Heisenberg antiferromagnet only increased with the discovery of cuprate high-T_c superconductivity as its origin was soon found to be connected to the ground state properties of the Heisenberg antiferromagnet [And87].

Recently, spin systems became relevant even for quantum optics, as it was theoretically shown [JBC$^+$98] and experimentally demonstrated [GME$^+$02] that systems of cold atoms trapped in an optical lattice can be tailored to realize the bosonic version of the Hubbard Hamiltonian. This opens the exciting possibility to use such setups as a kind of analog computer to simulate spin systems.

Up to now, however, we have to use digital computers, and the following section describes, rather briefly, the most important techniques commonly used for attacking problems on spin system. In the chapters following this one, our work on novel variational techniques for spin systems is described. The purpose of the present chapter is hence to give an overview of the state of the art in order to be able to rate the new techniques in the context of what can already be done with established ones. This overview does not aim to convey much more than the basic flavour of the described methods. Instead, I aim to provide a survey of introductions and reviews by suggesting selected literature.

6.1 Finite-size and infinite-size techniques

In statistical and solid-state physics, one is usually interested in the properties of very large systems; so large, in fact, that their properties are described by the so-called thermodynamic limit of infinite system size. Certain numerical techniques work natively in this limit, others estimate the numbers of interest for finite systems of different sizes, and then obtain the limit by extrapolation (usually done by extrapolating for the reciprocal system size tending to zero).

The possibility of such an extrapolation is due to the fact that from a certain order of magnitude for the system size onwards, the influence of the microscopic properties changes only quantitatively and slightly. Roughly speaking, this means: In a short spin chain, it may make a huge difference whether a spin has four or eight neighbors, as near neighbors may have a different effect than "very near" neighbours. For a long spin chain, the spins far away from a given one will all have roughly the same effect. Whether a spin "feels" five hundred or thousand neighbors in far distance can change values only slightly and smoothly. This allows to fit e. g. a power law or a polynomial and do a reliable extrapolation. The fitting works only, however, if the precision of the values obtained for different system sizes is sufficient to capture the possibly small differences between these. If a method allows to calculate for, say, one, two, and three hundred spins but with an uncertainty above the difference to be expected between these different sizes, any attempt to fit and extrapolate is bound to drown in noise. It is hence important to bear in mind that the ability of a technique to treat large systems is without value without good precision. Worse even, as the expected changes get smaller and smaller with system size, ever more precision is needed for larger systems.

In order to estimate the precision of an approximative method, exact values for comparisons are useful. These may be provided by exact diagonalisation of very small systems with the Lanczos algorithm [Lan50]. Modern implementations (ARPACK [LMSY96], SPINPACK [Sch]) make this feasible for up to roughly 25 spin-½ sites, which is sometimes even sufficient for a first rough estimate of thermodynamic limits.

6.2 Series expansions

In many systems, the approach towards critical points can be treated, despite high correlation, by means of series expansions, which become exact in the limit of inclusion of infinitely many terms. While a simple Taylor series often fails to have a sufficient radius of convergence to reach the parameter regions of interest, more sophisticated expansion techniques known from complex analysis, most importantly the Padé expansion [Bak61], have provided a wealth of results about the precise position of critical points and the values of critical exponents for a variety of spin systems. These expansion techniques work directly with infinite systems and can hence, wherever they are applicable, achieve estimates for thermodynamic limits at much higher precision than other techniques. As series expansion schemes are rather different from the ansatzes described in this thesis, I shall not say more about them but may direct the interested reader to the comprehensive monograph [Bak90]. Furthermore, the very recent result [BDL07] may be of specific interest in the context of the applications discussed here; it discusses a series expansion technique allowing for perturbative treatment of interaction Hamiltonians of arbitrary geometry provided that their strength is in a certain sense small compared to the spectral gap of the local Hamiltonian.

6.3 Real-space renormalization group

The concept of renormalization has been developed in quantum field theory to deal with the divergence of series expansions into Feynman diagrams and became, over several decades of research[2], a most fundamental methodological concept for any problem involving many different scales of length

---

[2] Of the many researchers who have contributed to renormalization, Wilson ranks Bethe, Schwinger, Feynman and Dyson as the most important ones. [Wil75]

or energy that cannot clearly be separated. This situation occurs not only in the evaluation of loop corrections in quantum electrodynamics, the original starting point, but also in many problems of solid state physics, and hence, the subject was soon recognized as a collaborate effort of high-energy and of solid-state physicists. Wilson's solution [Wil75] of the Kondo problem[3] is usually considered the most spectacular success in applying renormalization to a solid state setting. He introduced there the concept of the real-space renormalization group, which uses Kadanoff's idea of spin blocking (or: "block decimation") to move the high energy physics concepts from momentum to position space and then find the scaling relations for the Kondo setting. (These scaling relations, postulated by Widom, are the mathematical formulation behind the idea which is often called the universality of phase transitions.) Kadanoff's spin blocking is a way of coarse graining a lattice of spins: Briefly put, we may expect near a phase transition that the physics looks similar at all length scales because with the diverging correlation length a natural length scale is no longer present. Hence we may coarse-grain the infinite lattice by dividing it up into small, equal-sized blocks, each containing a few neighboring spins. Every block is then replaced by a single spin whose state is derived from the states of the spins in the block according to some chosen rule. The rescaled lattice, i. e. the lattice of these blocked spins, is then described by a Hamiltonian whose form is typically the same as before but whose coefficient have changed according to a relation that has to be derived. This renormalization relation is iterated to go to larger and larger blocks until it does not change any more. At such a fix-point we may assume, according to Kadanoff and Wilson, that all microscopic degrees of freedom have been "averaged out" and what is left describes the macroscopic appearance of the near-critical system. The fix-point form of the transformation then gives access to these properties and the critical exponents of the Widom scaling at this critical point.

This brief account may only give a very rough idea of what real-space renormalization is about. The interested reader may find a full exposition focusing on the Kondo problem in Wilson's review [Wil75], an account of renormalisation's history in Wilson's Nobel lecture [Wil93], a modern treatment (focusing on the Ising system as example) in Kadanoff's textbook [Kad00] and in Fisher's excellent review [Fis98], and a shorter overview focusing on computational aspects in Pang's textbook on computational physics [Pan97].

---

[3] Kondo noticed that certain metals with very dilute ferromagnetic impurities show peculiar behavior at very low temperature, namely the impurities' effect on, e. g., electrical resistance, is much larger than one should expect from the direct coupling of the impurities' spins. He conjectured the reason to be that the coupling of the impurity spins to the spins of conducting electrons becomes dominant below a temperature of the order of 1K. The problem got a notorious reputation of being extremely hard after many, very different, approaches failed to solve Kondo's deceptively simple looking model Hamiltonian for the zero-temperature limit of this situation. Wilson got his Nobel prize for demonstrating the power of his real-space renormalization group by finally solving the mystery of the Kondo problem.

The convoluted history of the field seems to make it especially hard for the newcomer to know where to start, and hence I also would like to recommend Domb's monograph [Dom96], which is simultaneously an introduction to and a chronicle of renormalization.

Wilson's original position-space renormalization technique is rarely used now. This is because, in startling contrast to the work on the Kondo problem, it turned out to fail for many systems of interest. Nevertheless, many aspects have been incorporated into other methods. Especially, it gave rise to the density-matrix renormalization group (DMRG), to be discussed in the next section.

Another fruitful approach was to merge Wilson renormalization with Monte Carlo integration: In the expression for the renormalization of the Hamiltonian, an average of the exponentiated inter-block part of Hamiltonian, weighted by the Boltzmann factors with respect to the intra-block part, has to be performed. For some systems, this average can be approximated well by means of a short Taylor expansion. More often, however, this Taylor series converges too slowly, and then, it is of huge advantage to use the Metropolis algorithm. This technique is known as Monte Carlo renormalization group and has been pioneered by Swendsen [Swe79]. As Kolb showed for the example of the Ising model with transverse field the method can also be used for quantum systems [Kol83].

Another modern renormalization algorithm that is based on Kadanoff–Wilson position space renormalization is contractor renormalization (CORE) [MW94, MW96]. Here, the fact is used that a Hamiltonian "contracts" a trial state towards the ground state if e^{-tH} is applied to it. The blocking scheme is now used to find a good approximation to the operator e^{-tH} (for t smaller than some $t_{\max}$) by projecting onto a set of low lying states within the Hilbert space of a block. This gives rise to an effective Hamiltonian, for which the real-space renormalization group is employed. The technique was first used semi-analytically and recently transformed into a fully numerical method [CLM04, SW06].

Finally, one should mention that many typical methods from quantum field theory owe their applicability to condensed matter theory to the work on the renormalization group. A very productive tool is here the so-called epsilon expansion, which treats systems with $d < 4$ dimensions as a perturbation of the four-dimensional case (which shows less divergence in path integral expansions), expanding around $\varepsilon = 4 - d$. (For more information, see e. g. [Dom96].)

6.4 Density-matrix renormalization group

6.4.1 Standard DMRG

Despite the impressive success that Wilson had with applying real-space renormalization to the Kondo problem, many subsequent attempt to use the method for other problems led to failures. In Ref. [WN92], White and Noack analyzed the reason for these failures. Following a suggestion by Wilson, they studied the discretised version of the standard textbook problem of a single quantum particle in a one-dimensional infinite potential well. They demonstrated that the failure of the method is due to the fact that the current block is unaware of its surrounding which forces unnatural boundary conditions onto the wave function. The solution proposed by White soon afterwards [Whi92, Whi93] was to apply renormalization not to wave functions but to density matrices, which are seen as result of tracing out the environment of the block from the pure wave function of the full, possibly infinite, system. This "density-matrix renormalization group" (DMRG) has turned out to be a very powerful technique to treat one-dimensional spin chains, because there, it is a natural approach to simply use a copy of the state of the block as a preliminary model for the environment.

The rationale behind White's DMRG is exposed very clearly in his review [Whi98], and I hence shall only mention the method's key points: We have an infinite chain of d-level systems, governed by a translationally invariant Hamiltonian[4] of the form $H_{\text{chain}} = \sum_a H^{(a,a+1)}$, and wish to approximate the expectation value of an observable O for the ground state (or a low-lying excited state). The method consist of two parts, often called the "warm-up" and the "sweeping phase". The first part, also denoted the "infinite size DMRG", starts with a four-site chain and builds up a longer and longer chain by inserting pairs of sites in the middle. Once a size has been reached which is considered sufficiently long, precision can be improved by "sweeping through" the chain and making local corrections.

In a nutshell, the warm-up works as follows: Start with a four-site chain (with open boundary conditions) and get its exact ground state by diagonalizing the Hamiltonian with the Lanczos algorithm. Now, trace out the right half (the "environment") in order to get a reduced density matrix ρ that describes the left half (the "block"). Diagonalize ρ and discard all but the eigenstates $|u_i\rangle$ corresponding to the D largest eigenvalues λ_i. The block is now approximated as $\rho \approx \sum_{i=0}^{D-1} \lambda_i |u_i\rangle \langle u_i|$. In most cases, the eigenvalues of ρ fall off very quickly so that for sufficiently large D the approximation is very good. Now, insert two further spins in the middle. The block and the left one of these spins form what is called the "superblock". States in the superblock's Hilbert space can now be approximated by projecting into the

---

[4] One can also treat more general Hamiltonian; especially the requirement for translational invariance can be dropped in a straight-forward manner.

subspace formed by the tensor product of the truncated block space spanned by $\{|u_i\rangle\}_{i=0}^{D-1}$ and the single-spin space $\mathbb{C}^d$. This subspace is isomorphic to $\mathbb{C}^D \otimes \mathbb{C}^d$, and the space of both the superblock and the "super-environment" is $\mathbb{C}^D \otimes \mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^D$. The Hamiltonian (or more precisely, the Hamiltonian truncated by the projection onto the $|u_i\rangle$) can hence be written as a $D^2d^2 \times D^2d^2$ matrix by using the transformation matrix formed by the $|u_i\rangle$s' expansion in the old basis. A Lanczos diagonalisation of the Hamiltonian yields a new ground state, now for the enlarged chain, but still living in the manageable D^2d^2-dimensional Hilbert space. Again, we trace out the right half of the now longer spin to get a new reduced density matrix ρ, and iterate this procedure until the chain is long enough.

Note that the ground state is always represented as a state in a truncated Hilbert space of the form $\mathbb{C}^D \otimes \mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^D$ (or, in the very first step, $\left(\mathbb{C}^d\right)^{\otimes 4}$). The outer parts, $\mathbb{C}^D$, always encompass the bulk of the chain, each of the D basis states representing a, typically highly entangled, state on one half (minus one site) of the chain. As the truncations are governed by the spectrum of the reduced density matrix, which captures the entanglement between the two halves of the chain, the "boundary" condition in the middle, where new sites are inserted, is typical for sites in the middle of the system, not for those at the actual boundary. This is the crucial feature that gives rise to the tremendous advantage that DMRG has over Wilson position space renormalization, where the state was always extended into the "void" of yet untreated length scales. However, the scheme relies on the idea of introducing single sites along the boundary. While in a chain these are only two sites, on a $L \times L$ plane, there would be $2L$ sites, blowing up the intermediate Hamiltonian to be diagonalized to a $D^2d^{2L} \times D^2d^{2L}$ matrix. Such a huge matrix can no longer be written down, let alone diagonalized. This is why DMRG is confined to one-dimensional systems.

In order to find the expectation value of an observable, the observable has to be represented by a $D^2d^2 \times D^2d^2$ matrix in the same fashion as the Hamiltonian, so that in the end it can be trivially applied to the resulting state. Note, by the way, that DMRG can also find low lying excited states, not just the ground state, by targeting not the lowest but one of the other low eigenvectors in the Lanczos diagonalisation.[5] This is a very useful feature that we shall miss in the variational methods to be discussed later.

After one has built up a state representing a chain of sufficient length, one may further improve the accuracy by doing a few so-called sweeps. In the warm-up one always inserts two sites between the two blocks and then absorbs them into the block. In a sweep step, one absorbs only one spin into its adjacent block and then, instead of inserting two new sites, separates off one site from the other block. The effect is that the chain keeps constant

---

[5] Remember that Lanczos diagonalisation yields only the lowest and the highest eigenvalues, not those in the middle.

length but the two individual spins move away from the middle towards the border. Each step involves a diagonalisation for the absorption of one spin into the growing block, which gives the chance to improve the precision of the calculation. This is because in the warm-up, a copy of the block serves as a place-holder for the environment to trace over. Now, we trace over a better approximation of the environment, and the chain converges very quickly to a fix-point, which usually is extremely close (with an energy value precise to five or more digits) to the correct state.

DMRG has been used for a large array of problems: the book [PWKH99] and the review [Sch05] give an overview over these diverse applications.

6.4.2 Improvements

When done in the style described above, the observables have to be transformed along with the state in order to be available in the truncated basis. Alternatively, one may store the operators that carry out the truncation. As Östlund and Rommer realized [ÖR95] the state's expansion in the product basis can then be written in the elegant form

$$|\Psi\rangle = \sum_{s_1,\ldots,s_N=0}^{d-1} \boldsymbol{\alpha}^{[s_1]} A^{2,[s_2]} A^{3,[s_3]} \ldots A^{N-1,[s_{N-1}]} \boldsymbol{\beta}^{[s_N]} \, |s_1 s_2 \ldots s_N\rangle \, , \tag{6.1}$$

where the $D \times D$ matrices $A^{a,[s]}$ contain the transformation that was found in the in the states $|u\rangle$ of the previous description. The vectors $\boldsymbol{\alpha}^{[s]}$ and $\boldsymbol{\beta}^{[s]}$ reflect the very first basis truncation and account for boundary conditions. In the case of periodic boundary condition, the from becomes even simpler, namely

$$|\Psi\rangle = \sum_{s_1,\ldots,s_N=0}^{d-1} \operatorname{tr}\left(A^{1,[s_1]} A^{2,[s_2]} \ldots A^{N,[s_N]} \right) |s_1 s_2 \ldots s_N\rangle \, , \tag{6.2}$$

These states are known as "matrix product states" (MPS) or "finitely-correlated states" and have already been studied previously by various authors. They turn out to be a generalization of the so-called valence-bond solid states (VBS), which are the solution of AKLT Hamiltonian (an exactly solvable model for one-dimensional antiferromagnets) [AKLT87]. Especially, Fannes et al. [FNW92] have determined that such states are capable of describing all states of a spin chain that are finitely correlated. (The precise meaning of this limitation is discussed later.) This made clear the limitations of DMRG, namely that it is incapable of representing spin chains whose correlations fall off slower that exponentially.

Östlund and Rommer used the MPS form to reformulate DMRG as a variational method: The general form (6.2) is seen as trial wave function and for an infinite translationally invariant chain, the ansatz can be simplified by making the matrices $A^{a,[s]}$ independent of the spin index a. Minimizing the energy with respect to a translationally-invariant Hamiltonian they could reproduce well known value for the general AKLT Hamiltonian [ÖR95, RÖ97].

This become the starting point for a variety of fruitful improvements and generalizations of DMRG. Verstraete et al. [VPC04] revisited the construction of matrix product states from the valence bond solid picture and realized that this solves an old problem of DMRG, namely its bad performance under periodic boundary conditions. Working consequently in the form (6.2) and using the eigenvalue minimization also used by Östlund and Rommer, but *without* making the matrices $A^{a,[s]}$ independent of a (not even for translation-invariant Hamiltonians) gives a performance as good as that of standard DMRG with open boundary conditions (surprisingly even though the warm-up phase was omitted).

As this ansatz will also be the basis of much of Chapter 9, we shall briefly sketch this technique using the formulation employed later: For a matrix product state $|\Psi\rangle$ of the form (6.2) and an observable O which is a tensor product of local observables, i. e.,

$$O = \bigotimes_{a=1}^{N} O_a,$$

the (unnormalized) expectation value of O can be computed as follows:

$$\begin{aligned} \langle \Psi \,|\, O \,|\, \Psi \rangle &= \sum_{\substack{s'_1,\ldots,s'_N,\\ s_1,\ldots,s_N=0}}^{d-1} \left(\prod_{a=1}^{N} \langle s'_a \,|\, O_a \,|\, s_a \rangle \right) \left(\operatorname{tr} \prod_{a=1}^{N} A^{[a],s'_a{}^*} \right) \left(\operatorname{tr} \prod_{a=1}^{N} A^{[a],s_a} \right) \\ &= \operatorname{tr} \prod_{a=1}^{N} \underbrace{\sum_{s',s=0}^{d-1} \langle s' \,|\, O_a \,|\, s \rangle \left(A^{[a],s'^*} \otimes A^{[a,s]} \right)}_{=:B^{[a]}} . \end{aligned} \tag{6.3}$$

We may rewrite the matrices $A^{[a],s}$ for a site a as a vector by concatenating the columns of the matrices,

$$\mathbf{a}^{[a]} = \bigoplus_{s=0}^{d-1} \bigoplus_{r=0}^{D-1} A^{[a],s}_{\cdot,r}.$$

Then, for a fixed site a, the expectation value is a quadratic form in $\mathbf{a}^{[a]}$, i. e.,

$$\langle \Psi \,|\, O \,|\, \Psi \rangle = \mathbf{a}^{[a]\dagger} \tilde{O}^{[a]} \mathbf{a}^{[a]}$$

with

$$\tilde{O}^{[a]}_{r's',sr} = \left(B^{[a+1]} \ldots B^{[N]} B^{[1]} \ldots B^{[a-1]} \right)_{r'r} \langle s' \,|\, O_a \,|\, s \rangle . \tag{6.4}$$

We now wish to find the optimal value for the matrices $A^{[a],s}$ at a certain site a such that the energy $E(\Psi) = \langle \Psi | H | \Psi \rangle \,/\, \langle \Psi | \Psi \rangle$ is minimal with respect to the values of the matrices at the other sites $b \neq a$ being held fixed. To do so we have to minimize the quotient

$$\frac{\mathbf{a}^{[a]\dagger}\tilde{H}^{[a]}\mathbf{a}^{[a]}}{\mathbf{a}^{[a]\dagger}\tilde{\mathbb{1}}^{[a]}\mathbf{a}^{[a]}},$$

where the tilde on an operator denotes the forming of a matrix according to Eq. (6.4). (For this, the Hamiltonian has to be written as a sum of tensor product observables, $H = \sum_i H_i$, each of which forms a matrix, and these matrices are summed up again, $\tilde{H}^{[a]} = \sum_i \tilde{H}_i^{[a]}$.)

Such a quotient of two quadratic forms is known as a generalized Rayleigh quotient and can be minimized by assigning to $\mathbf{a}^{[a]}$ the generalized eigenvector corresponding to the smallest generalized eigenvalue of the generalized eigenvalue problem $\tilde{H}\mathbf{a} = \lambda\tilde{\mathbb{1}}\mathbf{a}$. Using the QZ algorithm [MS73] (an implementation of which is provided in LAPACK [ABB^{+}99]), this vector can be found efficiently and accurately.[6]

6.4.3 Time evolution

Various attempts have been undertaken to study dynamics and time evolution with DMRG. When staying within White's formalism, one has to be careful to allow for proper adaption of the bases. This mistake was made in the attempt of Ref. [CM02] and identified in the comment [LXW03], where a resolution was also suggested. Due to the limited space in a PRL comment, a proper exposition of the method was not done there but may be found in Refs. [MMN05, Man06].

Another approach came from the quantum information community: In order to contribute to the then very lively debate about what makes a quantum computer powerful, Vidal argued that strong entanglement is crucial, because a quantum computation which leaves the entanglement weak can be simulated efficiently on a classical computer. Vidal proved this point by giving an explicit algorithm to do so [Vid03]. Weak entanglement here means that if one cuts the quantum register in two chains, then the Schmidt rank of the state with respect to this bipartition has to stay bounded by a constant. This is because Vidal's representation, which allows to read off the Schmidt representation for each such bipartition, is very similar to the matrix product state representation and hence has the same limitations on representable states. Ref. [Vid04] discusses how the algorithm (now called time-evolving block decimation (TEBD)) may be used to simulate the time evolution of a spin chain and makes the connection to DMRG theory. Further elaborations and applications of TEBD are discussed in Refs. [DKSV04, WF04].

---

[6] As $\tilde{H}$ and $\tilde{\mathbb{1}}$ are Hermitean, one may alternatively reduce the problem to an ordinary eigenvalue problem by means of a Cholesky factorization of $\tilde{\mathbb{1}}$ [TB97, ABB^{+}99].

6.4.4 Generalizations to other geometries

The restriction of DMRG to chains and rings is a severe limitation, and generalizations to geometries at higher dimensions were always sought for. A simple scheme is to look at ladders, i. e., pseudo-two-dimensional lattices on long strips that are only a few sites wide. There, each DMRG spin represents the (typically two to four) spins forming one step of the ladder. This is useful e. g. in order to study models for frustrated systems.

Attempts to use the chain-like matrix product states to treat 2D systems by letting the spin chain run through the lattice in "serpentines" have been not very successful. This is not surprising as the strong correlations between nearest neighbors are mediated by the many matrices between a spin in one row and the next. Systems with tree-like geometries, however, still allow for treatment in this manner (as will be seen in Chapter 9) and this has been used successfully to study exciton transport on tree-shaped molecules [MRS02, Rod02].

In Ref. [VC04b], Verstraete and Cirac revisit the valence bond construction already mentioned above (see also Sec. 5.2.1) and study the entanglement properties of states so gained. This led them to a natural extension of matrix product states now called projected entangled-pair states (PEPS) [VC04a]. These states may be formed to accommodate any lattice geometry (or even any graph), have entanglement properties that may keep up with the area law (to be discussed in detail in Sec. 9.4), and reduce, when constructed for a chain geometry, to the matrix product states. An algorithm for evaluation of observables is provided (which, however, needs to do certain approximations), and ground states of given system Hamiltonians may be found by performing imaginary time evolution with a straight-forward generalization of TEBD. The recent article [MVC07] show how the method works in practice. I shall not go into more details here but postpone the discussion to Chapter 9, where we discuss PEPS in a more general context.

Finally, it is worth mentioning that by means of transfer matrices, spin-chain DMRG can also treat 2D classical systems [Nis95], which in turn allows (via the Suzuki mapping, see Sec. 6.5) to study thermal properties of 1D quantum systems [BXG96]. An alternative ansatz, based on the introduction of mixed state analogs to the pure MPSs, is explored in Refs. [VGC04, ZV04]. It is superior to the transfer matrix method as it does not require translation invariance and works better for finite-size systems and at low temperature.

6.5 Quantum Monte Carlo

The idea that randomness can help to numerically solve exact problems more efficiently, is not new.[7] Its great potential has been realized by Ulam during

---

[7] Buffon's needle (1733) [Wei05] may be one of the oldest examples though it is not exactly more efficient than a non-randomized calculation.

the Manhattan project, who coined the term "Monte Carlo integration" for the use of the fact that the precision of numerical integration falls of exponentially with the number of function arguments when integrating along a regular grid but stays good when sampling a randomly chosen points. The idea got boosted by the discovery of the Metropolis algorithm [MRR+53], which allows to study thermal equilibrium states of classical systems (and, as Hastings realized [Has70], also statistical systems far from physics). This is now known as "Metropolis-Hastings algorithm" or "Markov chain Monte Carlo". In essence, the Metropolis algorithm is a scheme to sample the phase space (or solution space) according to a probability distribution whose normalization cannot be calculated by choosing the sample points with a random walk biased with the so-called Metropolis condition (also known as the "detailed balance"). A good introduction to this topic can be found in the classic little book of Hammersley and Handscomb [HH64].

Suzuki's seminal observation that d-dimensional quantum spin systems may be mapped onto $(d+1)$-dimensional classical systems [Suz76] paved the way to quantum Monte Carlo (QMC) [SMK77]. The idea of the Suzuki mapping is quickly explained: In order to find a value of interest, say, the partition function $Z = \operatorname{tr} e^{\beta H}$ of a system of N spins (each with n levels) with Hamiltonian H at inverse temperature β, we use the product basis $\mathcal{B}$ of $(\mathbb{C}^n)^{\otimes N}$, and write

$$Z = \operatorname{tr} e^{-\beta H} = \lim_{\substack{L\to\infty \\ \Delta\beta=\beta/L}} \sum_{\mathbf{s}_1\in\mathcal{B}} \langle \mathbf{s}_1 | \left(\mathbb{1} - \Delta\beta H\right)^L | \mathbf{s}_1 \rangle \,.$$

Inserting decompositions of the identity before each factor of the Trotter decomposition, we get

$$Z = \sum_{\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_L\in\mathcal{B}} \langle \mathbf{s}_1 \,|\, (\mathbb{1} - \Delta\beta H) \,|\, \mathbf{s}_2 \rangle \langle \mathbf{s}_2 \,|\, (\mathbb{1} - \Delta\beta H) \,|\, \mathbf{s}_3 \rangle \cdots \times$$
$$\times \ldots \langle \mathbf{s}_L \,|\, (\mathbb{1} - \Delta\beta H) \,|\, \mathbf{s}_1 \rangle \,.$$

This is now formally identical to the partition function of a *classical* system with $N \times L$ spins, which is then treated with the Metropolis algorithm. Using a large N brings us towards the thermodynamic limit of large systems, and a large L towards the limit of zero temperature.

Nowadays, the term "quantum Monte Carlo" (QMC) is used for a variety of techniques. These include on the one hand techniques for thermal states at finite temperature, namely path-integral Monte Carlo (Suzuki et al.'s initial method), world-line Monte Carlo and stochastic series expansion (refinements of it), and on the other hand techniques to find the $T = 0$ ground state, namely variational Monte Carlo and Green's function Monte Carlo. Variational Monte Carlo is a variational method using "classical" Monte Carlo integration to evaluate the energy of a given parametrized trial wave function. We shall briefly describe it in Sec. 6.6. Green's function Monte Carlo uses imaginary time evolution to approximate a ground state. It achieves an astounding accuracy at quite modest computational effort: Already in 1989, the ground state properties of the Heisenberg antiferromagnet were calculated, reaching a precision of more than 3 digits for the energy at the thermodynamic limit, with the calculation time for a 12 × 12 lattice being mere 3 hours on a supercomputer[8] [TC89]. Three more digits of precision, in the energy as well as in other quantities, has been achieved only shortly afterwards [Run92].

A current overview on the use of Monte Carlo techniques throughout physics may be found in the book [LB05], which, however, restricts its treatment of QMC to only a brief one-chapter overview over the just mentioned terms. Of these techniques, the modern variants of path-integral Monte Carlo currently seem to enjoy most interest of the community. This is mainly due to the spectacular overcoming of the so-called "critical slowing down" problem, which prevented the application of path-integral Monte Carlo to critical or near-critical systems due to exponentially increasing convergence time. (Also, these methods now reach very low temperatures, allowing them to outperform dedicated $T = 0$ techniques in ground state studies.) The problem of critical slowing down was overcome by adapting the solution for the corresponding problem in classical Markov chain Monte Carlo (the so-called cluster-flipping scheme [SW87]) by two clever constructions known as the loop algorithm [ELM93] and the worm algorithm [PST98]. A review of these inventions and their applications may be found in [KH04]. (Unfortunately, Ref. [KH04] assumes the reader's familiarity with path-integral QMC as is was done before the loop algorithm, which may be obtained from the older reviews [RL85] or [RL92].) Alternatively, a reader in hurry may find an accessible and qualified, but quite short introduction, in part II of the thesis [Weß05].[9] The book [NU99] may also provide an overview on the variety of modern quantum Monte Carlo schemes.

The ALPS project [ADG$^+$05, AAC$^+$07] is a recent undertaking of offering —in a consistent framework— modern implementations for various path-integral Monte Carlo techniques (including the worm, loop, or directed-loop algorithm), as well as some non-Monte-Carlo schemes. This software package allows also the non-expert physicist to experiment with these techniques. The size of systems treatable with path-integral Monte Carlo is impressive, as can be seen from some examples: Ref. [WATB04] studies the co-existence of superfluid and Mott phases in a boson Hubbard gas in a optical lattice of 50 × 50 sites with inhomogeneous trapping potential, and achieves excellent

---

[8] Note that the supercomputer that they used then achieved less than a fifth of the performance of a standard PC of today.

[9] These references chiefly treat the case of systems defined on lattices, which is what we are concerned with in this part of the thesis. Quantum Monte Carlo techniques for continuous systems —most successfully used for liquid helium near the superfluidity transition— is reviewed e. g. in [Cep95].

precision. The 2D Heisenberg antiferromagnet, already mentioned earlier as a standard example for a difficult spin system, has been treated for up to 200×200 spins with a precision suitable to get robust and precise extrapolations to the thermodynamic limit [CRT$^+$03].

QMC simulations of dynamical evolution in time have not yet progressed as far, but there are algorithms such as, e. g., those used in Refs. [EM94, OP05], which seem promising.

The major draw-back of all Monte Carlo techniques is the so-called sign problem: The precision of Monte Carlo integration breaks down dramatically if the integrand cannot be written with a positive probability density to govern the Metropolis walk. In the case of quantum systems, this means that due to their non-vanishing anticommutator, fermionic systems can only be treated in special cases, and the treatment of frustrated systems is severely hindered as well. This renders path-integral Monte Carlo calculations for such systems almost impossible and variational Monte Carlo calculation are much less accurate as otherwise. (See [TW05] for a recent analysis of the difficulties in overcoming this old problem.) For an example on how far one gets despite this problem, see, e. g., newer work on the Heisenberg antiferromagnet on frustrated lattices (e. g., [CTS00]) or on the fermionic Hubbard model (e. g., [FKK06]).

6.6 Variational methods

The variational method is not so much a method in its own right, but rather a common theme which we find in many of the numerical techniques (and hence we have already come across it above several times in this chapter). The basic idea is to use a "trial wave function" (or "test function") which depends on a set of K real parameters, i. e. is a map from $\mathbb{R}^K$ into the Hilbert space of the system to be studied. As K is much smaller than the dimension of the Hilbert space (necessarily, as otherwise it would be too expensive to determine all K parameters), the manifold mapped out by these states is only a very small part of the Hilbert space. Only a good choice of the map allows to find a state that is close to the real ground state. To this end, the energy expectation value is written as a function of the K parameters, and this function is minimized. In Sec. 8.2, we shall come back to the question under which conditions such a programme can succeed.

As the real ground state usually cannot be expected to lie in the manifold of trial wave functions, one is bound to always overestimate the energy, leading to an unavoidable source of systematic error. This is in contrast to methods such as path-integral Monte Carlo, which are exact in the limit of infinite sample size and a controlled approximation, i. e., the uncertainty of the result can be estimated reliably.

Many of the successful choices for trial wave functions are based on the Jastrow wave function [Jas55]. Further common choices, both ultimately based on Jastrow's work, are the Gutzwiller approximation [Gut63] (reviewed in [Vol84]) and the resonating valence-bond states (see, e. g., [LDA88]).

For simple cases, the expression for the energy as function of the trial state parameters can be minimized analytically or with exact numerics. Matrix product states and their variants discussed in Sec. 6.4 are instance of rather sophisticated trial functions that still allow an exact evaluation, or a good approximation (in case of PEPS states) of their energy. Another possibility is the variational Monte Carlo (VMC) technique. There, the energy is an integral or sum over the basis set or configuration space and is evaluated with the Metropolis algorithm. This was first demonstrated in a calculation for liquid helium [McM65], and later shown to give good results even for fermionic systems [CCK77, YS87].

A very recent idea, dubbed "string states", is to combine these approaches: VMC is not used, as usual, with Jastrow-type or similar wave function, but rather with variants of matrix product states augmented such that they have entanglement properties favorable for 2D or 3D geometries for the price that an evaluation of their energy expectation value is only possible with Metropolis sampling [SWVC07b] (see also [SV07]). In this work, a clever strategy for sample re-use is devised that allows to still use the genrealized-eigenvalue technique as described in Sec. 6.4.2 with its good convergence properties.

From the viewpoint of quantum information theory, an interesting alternative way to look at variation emerges, suggested in [DEO07]: Instead of minimizing within the manifold of variational states, one looks at the unitary operations on the Hilbert space that can map a fixed trivial initial state to a trial state. One chooses a class of unitaries that may be decomposed into a quantum-computational circuit consisting of quantum gates, and then uses knowledge from the theory of quantum computation and also of optimal control theory to find a good algorithm to optimize the circuit in order to yield a state of minimal energy.

In the following two chapters we present and explore a novel class of variational states, which are based on a generalization of the graph states that we have used in Part I. This generalization, called *weighted graph states*, is based on the application of commuting unitary operators on all pairs of spins of a product state, and the phases in these operators, as well as the parametrization of the product state, is the set of parameters that is varied to minimize the energy.

Chapter 7

PUBLICATION:[1] Ground state approximation for strongly interacting systems in arbitrary dimension

Simon Anders, Martin B. Plenio, Wolfgang Dür,
Frank Verstraete, and Hans J. Briegel

published in *Physical Review Letters, Vol.* ***97*** *(2006), 107206*
submitted 28 Feb 2006, published 8 Sep 2006
DOI: 10.1103/PhysRevA.PhysRevLett.97.107206

ArXiv preprint: quant-ph/0602230

Abstract

We introduce a variational method for the approximation of ground states of strongly interacting spin systems in arbitrary geometries and spatial dimensions. The approach is based on weighted graph states and superpositions thereof. These states allow for the efficient computation of all local observables (e. g. energy) and include states with diverging correlation length and unbounded multi-particle entanglement. As a demonstration we apply our approach to the Ising model on 1D, 2D and 3D square-lattices. We also present generalizations to higher spins and continuous-variable systems, which allows for the investigation of lattice field theories.

---

[1] This publication and the paper of Chapter 8 are reprinted in chronological order. The reader may, however, find the matter more accessible if (s)he reads Chapter 8 before the present one. (This one, being in letter format, is very dense, while the other one explains the same subject in a more comprehensive and extended style.)

[PACS: 03.67.Mn, 02.70.-c, 75.40.Mg, 75.10.Jm]

7.1 Introduction

Strongly correlated quantum systems are of central interest in several areas of physics. Exotic materials such as high-T_c superconductors and quantum magnets exhibit their remarkable properties due to strong quantum correlations, and experimental breakthroughs with e.g. atomic gases in optical lattices provide a perfect playground for probing strongly correlated quantum systems. The main obstacle in understanding the behavior of those quantum systems is the difficulty in simulating the effective Hamiltonians that describe their properties. In most cases, the strong correlations in the exponentially large Hilbert space render an exact solution infeasible, and attacking the problem by numerical means requires sophisticated techniques such as quantum Monte Carlo (QMC) methods or the density matrix renormalization group (DMRG) approach [Whi92, Whi93, Sch05].

QMC methods suffer from the sign problem which makes them inappropriate for the description of fermionic and frustrated quantum systems. DMRG is a variational approach that provides approximations to ground states, thermal states and dynamics of many-body systems. Recent insight from entanglement theory have lead to an improved understanding of both the success and the limitations of this approach. Indeed, the accuracy of the method is closely linked to the amount of entanglement in the approximated states [Vid03, Vid04, VPC04]. Matrix product states [FNW92], which provide the structure underlying DMRG, are essentially one-dimensional and the entanglement entropy of these states is limited by the dimension D of the matrices, which in turn is directly linked to the computational cost [Vid03, Vid04, Sch05]. Hence a successful treatment of systems with bounded entanglement, e.g. one-dimensional, non-critical spin systems with short range interactions, is possible, while the method is inefficient for systems with an unbounded amount of entanglement, e.g. critical systems and systems in two or more dimensions. Promising generalizations that can deal with higher dimensional systems have been reported recently [VC04a, Vid05]. However, the computational effort and complexity increases with the dimension of the system. In addition, the amount of block-wise entanglement of the states used in Ref. [VC04a] still scales proportional at most to the surface of a block of spins, whereas in general a scaling in proportion to the volume of the block is possible. Such a scaling can in fact be observed for disordered systems [CHDB05] or systems with long-range interactions [DHH$^+$05].

Here we introduce a new variational method using states with intrinsic long-range entanglement and no bias towards a geometry to overcome these limitations. We first illustrate our methods for spin-1/2 systems, and then generalize them to arbitrary spins and infinite dimensional systems such as harmonic oscillators. In finite dimensions, the method is based on a certain class of multiparticle-entangled spin states, *weighted graph states* (WGS) and superpositions thereof. WGS are a $O(N^2)$ parameter family of N–spin states with the following properties: (i) they form an (overcomplete) basis, i.e. any state can be represented as a superpositions of WGS; (ii) one can efficiently calculate expectation values of any localized observable, including energy, for any WGS; (iii) they correspond to weighted graphs which are independent of the geometry and hence adaptable to arbitrary geometries and spatial dimensions; (iv) the amount of entanglement contained in WGS may be arbitrarily high, in the sense that the entanglement between any block of N_A particles and the remaining system may be $O(N_A)$ and the correlation length may diverge.

Note that (iii) and (iv) are key properties in which this approach differs from DMRG and its generalizations and which suggest a potential for enhanced performance at least in certain situations, while (ii) is necessary to efficiently perform variations over this family. In the following we will outline how we use superpositions of a small number of WGS as variational ansatz states to find approximations to ground states of strongly interacting spin systems in arbitrary spatial dimension.

7.2 Properties of WGS

WGS are defined as states of N spin-1/2 (or qubits), that result from applying phase gates $U_{ab}(\varphi_{ab}) = \mathrm{diag}(1,1,1,e^{-i\varphi_{ab}})$ onto each pair of qubits $a, b \in \{1, 2, \ldots, N\}$ of a tensor product of σ_x-eigenstates $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, followed by a single-qubit filtering operation $D_a = \mathrm{diag}(1, e^{d_a})$, $d_a \in \mathbb{C}$ and a general unitary operation U_a

$$|\Psi_{\Gamma,\mathbf{d},U}\rangle \propto \prod_{a=1}^{N} U_a D_a \prod_{b=a+1}^{N} U_{ab}(\varphi_{ab})|+\rangle^{\otimes N}. \tag{7.1}$$

The phases φ_{ab} can be associated with a weighted graph with a real symmetric adjacency matrix $\Gamma_{ab} = \varphi_{ab}$. For convenience, we define a deformation vector $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ and $U \equiv \bigotimes_a U_a$. The deformations make WGS as used in this letter slightly more general than the WGS used in Refs. [CHDB05, DHH+05] where $d_a = 0$. One can conveniently rewrite $|\Psi_{\Gamma,\mathbf{d},U}\rangle$ as

$$|\Psi_{\Gamma,\mathbf{d},U}\rangle \propto U \sum_{s} e^{-is^T\Gamma s/2+\mathbf{d}^T s}|s\rangle, \tag{7.2}$$

where the sum runs over all computational basis states, which are labelled with the binary vector $s = (s_1, s_2, \ldots, s_N)^T$. Our class of variation states comprises superpositions of WGS of the form

$$|\Psi\rangle \propto \sum_{i=1}^{m} \alpha_i |\Psi_{\Gamma,\mathbf{d}^{(i)},U}\rangle, \tag{7.3}$$

i. e. the superposed states differ only in their deformation vector $\mathbf{d}^{(i)}$, while the adjacency matrix Γ and the unitary U are fixed. Such a state is specified by $N(N-1)/2 + 3N + 2(N+1)m = O(N^2)$ real parameters.

We now proceed to verify the properties set out in the introduction. For property (i), observe that for any fixed Γ and U, all possible combinations of $D_a \in \{\sigma_z^{(a)}, \mathbb{1}^{(a)}\}$ lead to an orthonormal basis (note that $\sigma_z^{(a)}, \mathbb{1}^{(a)}$ commute with U_{ab}). Hence *any* state $|\Psi\rangle$ can be written in the form Eq. (7.3) for sufficiently large $m \leq 2^N$, which shows the exhaustiveness of the description.

The relevance of employing deformations lies in the observation that only $|\Psi\rangle$ of the form of Eq. (7.3) permit the efficient evaluation of the expectation values of localized observables A, i.e. satisfy property (ii). For simplicity we restrict our attention to observables of the form

$$A = \sum_{a<b} A_{ab} + \sum_{a} A_a, \tag{7.4}$$

where A_{ab} has support on the two spins a, b. The method can be easily adopted to any observable that is a sum of terms with bounded support. To compute $\mathrm{tr}(A|\Psi\rangle\langle\Psi|) = \sum_{a<b} \mathrm{tr}(A_{ab}\rho_{ab}) + \sum_a \mathrm{tr}(A_a \rho_a)$ it is sufficient to determine the reduced density operators ρ_{ab} and ρ_a.

For a single WGS ($m = 1$)we obtain $\rho_{12} = (U_1 \otimes U_2)(\sum r_{\mathbf{s},\mathbf{t}} |\mathbf{s}\rangle\langle\mathbf{t}|)(U_1 \otimes U_2)^\dagger$ with

$$r_{\mathbf{s},\mathbf{t}} = e^{-i\gamma} \prod_{c=3}^{N} \left(1 + e^{d_c + d_c^* - i\sum_{e=1}^{2}(s_e - t_e)\Gamma_{ec}}\right) \tag{7.5}$$

and $\gamma = \sum_{a,b=1}^{2} \Gamma_{ab}(s_a s_b - t_a t_b) + \sum_{a=1}^{2}(d_a s_a + d_a^* t_a)$. This generalizes the formula for WGS without deformation obtained in Ref. [DHH$^+$05]. Eq. (7.5) demonstrates that for any WGS, the reduced density operator of two (and one) spins can be calculated with a number of operations that is linear in the system size N, as opposed to an exponential cost for a general state.

A straight-forward generalization of Eq. (7.5) allows one to calculate two–qubit reduced density matrices for superpositions of the form of Eq. (7.3) in time $O(m^2 N)$. Therefore the expectation value of an observable A of the form of Eq. (7.4) with K terms requires $O(m^2 K N)$ steps. This implies that even for Hamiltonians where all spins interact pairwise (and randomly), i.e. $K = N(N-1)/2$, the expectation value of the energy for our ansatz states can be obtained in $O(m^2 N^3)$ steps. For short–range interaction Hamiltonians, this reduces to $O(m^2 N^2)$. The total number of parameters (and memory cost) scales as $O(N^2 + mN)$, which can be further reduced by employing symmetries.

The adjacency matrix Γ, containing the interaction phases φ_{ab}, reflects the entanglement properties and the geometry of the system. For instance, a state corresponding to a linear cluster state [BR01], will have only $\Gamma_{a,a+1} \neq 0$, while $\Gamma_{a,a+l} \neq 0$ would correspond to longer-ranged correlations. Different values of φ_{ab} lead to very different (entanglement) properties: For $\varphi_{ab} = |\mathbf{x}_a - \mathbf{x}_b|^{-\beta}$, where $\mathbf{x}_a$ denotes the spatial coordinates of spin a, one obtains states with diverging correlation length for two-point correlations, while block–wise entanglement can either be bounded or grow unboundedly, depending on the value of β [DHH$^+$05]. As Γ may have arbitrary structure it can reflect also complicated geometries on lattices in higher spatial dimensions.

7.3 Variational method

Any state of the form Eq. (7.3) with $m = \mathrm{poly}(N)$ permits the efficient calculation of expectation values of any two–body Hamiltonian H. A good approximation to the ground state is then obtained by numerical optimization of the parameters characterizing the state, i.e. the $O(N^2 + Nm)$ real numbers describing Γ, U, α_j, and $\mathbf{d}^{(j)}$. Starting from random values, one descends to the nearest energy minimum using a general local minimizer (we used L-BFGS [BLN95]). Another approach that we found to work well is to keep all parameters fixed except for either those corresponding to (i) one local unitary U_a, (ii) one phase gate $U_{ab}(\varphi_{ab})$ or (iii) the deformation vector $d_a^{(j)}$ for one site a. In each case, the energy as a function of this subset of parameters turns out to be a quotient of quadratic forms, which can be optimized using the generalized-eigenvalue (Rayleigh) method. A similar result holds for the superposition coefficients α_j. One then optimizes with respect to these subsets of parameters in turns until convergence is achieved. If one increases m stepwise, one —somewhat surprisingly— does not get stuck in local minima.

A significant reduction of the number of parameters and the computational costs may be achieved by exploiting symmetries, or by adapting Γ to reflect the geometrical situation. For instance, for systems with short range interactions and finite correlation length, one might restrict the range of the weighted graph, i.e. $\Gamma_{ab} = 0$ if $|\mathbf{x}_a - \mathbf{x}_b| \geq r_0$. This reduces the number of parameters describing the WGS from $O(N^2)$ to $O(N)$. For translationally invariant Hamiltonians, a better scheme is to let Γ_{ab} depend only on $|\mathbf{x}_a - \mathbf{x}_b|$. This reduces the number of parameters to $O(N)$ as well, and it seems to hardly affect the accuracy of the ground state approximation. Hence, it allows one to reach high numbers of spins N and thus to study also 2D and 3D systems of significant size. Trading accuracy for high speed one may even use a fully translation-invariant ansatz, where also D_a and U_a are constant and independent of a. In the latter case, for Hamiltonians with only nearest-neighbor interactions, the expectation value of the energy can be obtained by calculating only a *single* reduced density operator, and the computational cost to treat 2D [and 3D] systems of size $N = L^2$ [$N = L^3$] turns out to be of $O(L)$ rather than $O(N)$.

7.4 Demonstration: The Ising model

Our method allows us to determine, with only moderate computational cost, an upper bound on the ground state energy of a strongly interacting system of arbitrary geometry. Together with the Anderson lower bound, one can hence obtain a quite narrow interval for the ground state energy and observe qualitative features of the ground state.[2] To illustrate our method, we have applied it to the Ising model in 1D, 2D and 3D with periodic boundary conditions, described by the Hamiltonian

$$H = -\sum_{\langle a,b\rangle} \sigma_z^{(a)}\sigma_z^{(b)} - B\sum_a \sigma_x^{(a)} \tag{7.6}$$

where $\langle a, b\rangle$ denotes nearest neighbors. For a spin chain with $N = 20$, and a 2D lattice of size 4×4 we compared our numerical ground state approximation with exact results (Fig. 7.1a). We have also performed calculations for larger 2D systems up to 14×14. We note that the accuracy can be further improved by increasing m (see Fig. 7.1b). In fact our numerical results suggest an exponential improvement with m. We have also tested the fully translation invariant ansatz with distance dependent phases, constant d_a and alternating U_a for 1D, 2D and 3D systems of size $N = 30$, $N = 900$ and $N = 27000$ respectively (see Fig. 7.2). There, for lack of a reference value for the exact ground state, we compare with the Anderson bound obtained by calculating the exact ground state energy E_A for system size $N = 15, 3^2, 2^3$ respectively. In the 2D and especially the 3D case it is not expected that the Anderson bound is particularly tight and may lead to a significantly underestimation of the precisions achieved by our approach. The states approximated with this simple ansatz also show qualitatively essential features of the exact ground state. As an example, the maximal two-point correlation function $Q_{\max}^{a,a+1}$ (where the two point correlation functions are defined as $Q_{\alpha,\beta}^{a,b} = \langle\sigma_\alpha^{(a)}\sigma_\beta^{(b)}\rangle - \langle\sigma_\alpha^{(a)}\rangle\langle\sigma_\beta^{(b)}\rangle$) is plotted against the magnetic field B in Fig. 7.2b. Strong indication for the occurrence of a phase transition can be observed: the correlations significantly increase around $B \approx 1.1, 3.12, 5.22$ in 1D, 2D, 3D respectively. This is in good agreement with estimates employing sophisticated power series expansions for the *infinite* systems or Padé approximants based on large scale numerical simulations, which expect the critical points at $B = 1, 3.04, 5.14$ [HHO90, WOH94]. We also remark that the approximated states show a scaling of block-wise entanglement proportional to the surface of the block, i. e. $S_{N_A} \approx \beta_B L^{\mathrm{dim}-1}$, where β_B is some constant depending on magnetic field B, $N_A = L^{\mathrm{dim}}$ and dim is the spatial dimension. We can estimate β_B and find that it significantly increases near the critical point.

---

[2] We remark that for 1D systems, the accuracies appear to scale less well in the resources as for DMRG methods. However, our approach yields accurate results also for 2D and 3D systems.

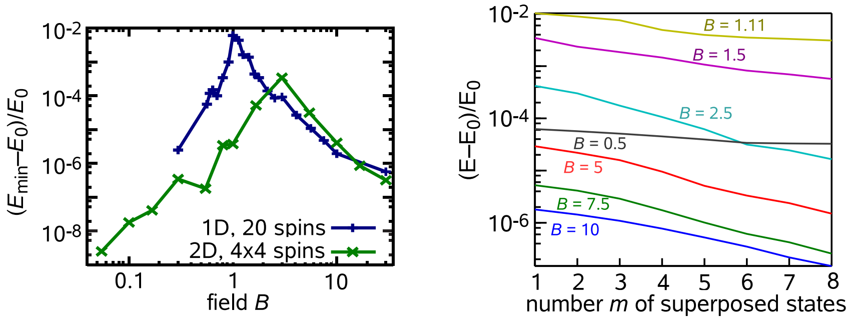


Figure 7.1: (Color online.) (a) Relative deviation from exact ground state energy for Ising chain with $N = 20$ (blue) and 4×4 2D lattice (green) with periodic boundary conditions as function of magnetic field B (calculated using BFGS minimization with summarised phases, $m \leq 6$). (b) 1D Ising chain with $N = 20$. Improvement of relative deviation from ground state energy as function of number of superposed states m for various field values B (calculated using Rayleigh minimization without summarised phases).

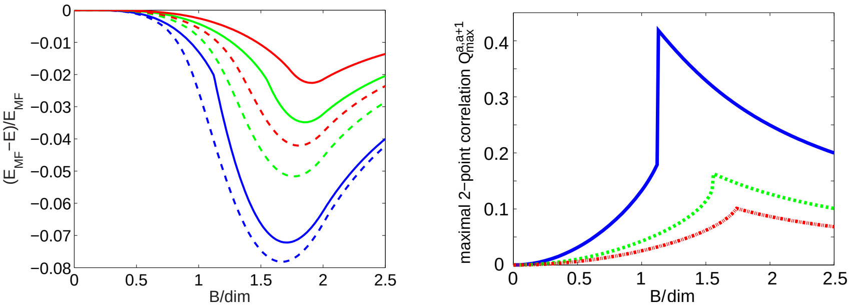


Figure 7.2: (Color online.) Ising model in 1D (blue) with $N = 30$, 2D (green) with $N = 30 \times 30 = 900$ and 3D (red) with $N = 30 \times 30 \times 30 = 27000$ spins arranged as chain, square, and cubic lattice, respectively, for fully symmetric ansatz states with $\varphi_{ab} = f(|x_a - x_b|)$, $d_a = 1$ as function of magnetic field B/dim, where dim is dimension of lattice. (a) Relative deviation of ground state energy $(E_{MF} - E)/E_{MF}$ per bond from to mean field approximation E_{MF} (solid), and of Anderson bound $(E_{MF} - E_A)/E_{MF}$ (dashed). Translational invariance is reduced by using $U_1 \neq U_2$ (alternating). (b) maximal two-point correlation $Q_{\max}^{a,a+1}$ for nearest neighbors.

7.5 Generalizations

Our approach can be adapted directly to spin-$\frac{n}{2}$ systems using the representation Eq. (7.2). There the sum over binary vectors $\mathbf{s}$ with $s_i = 0, 1$ has to be changed to n-ary vectors $\mathbf{s}$ with $s_i = 0, 1, ..., n-1$ and the corresponding matrices/vectors $\Gamma, \mathbf{d}, U$ have to be modified accordingly. However, the limit $n \to \infty$ to infinite dimensional systems is both problematic and impractical, as the computational effort increases with n. For continuous variable systems we thus choose a closely related but slightly different approach.

The description of field theories on lattices generally leads to infinite-dimensional subsystems such as harmonic oscillators. A Klein-Gordon field on a lattice for example possesses a Hamiltonian quadratic in position and momentum operators X and P whose ground state is Gaussian [AEPW02, PEDC05]. This suggests that techniques from the theory of Gaussian state entanglement (see [EP03] for more details) provide the most natural setting for these problems. To this end, consider N harmonic oscillators and the vector, $\mathbf{R} = (R_1, ..., R_{2N})^T = (X_1, P_1, ..., X_N, P_N)^T$. The canonical commutation relations then take the form $[R_j, R_k] = i\sigma_{jk}$ with the symplectic matrix σ. All information contained in a quantum state ρ can then be expressed equivalently in terms of the characteristic function $\chi_\rho(\boldsymbol{\xi}) = \mathrm{tr}[\rho W(\boldsymbol{\xi})]$ where $\boldsymbol{\xi} \in \mathbb{R}^{2N}$ and $W(\boldsymbol{\xi}) = \exp(i\boldsymbol{\xi}^T\sigma\mathbf{R})$. Then, expectation values of polynomials of X and P can be obtained as derivatives of χ. For Gaussian states, i.e. states whose characteristic function is a Gaussian $\chi_\rho(\boldsymbol{\xi}) = \chi_\rho(0)e^{-\frac{1}{4}\boldsymbol{\xi}^T\gamma\boldsymbol{\xi}+\mathbf{D}^T\boldsymbol{\xi}}$, where here γ is a $2N \times 2N$-matrix and $\mathbf{D} \in \mathbb{R}^{2N}$ is a vector, these expectation values can be expressed efficiently as polynomials in γ and $\mathbf{D}$. On the level of wave functions a pure Gaussian state is given by $|F, G; \mathbf{a}\rangle = C\int_{\mathbb{R}^N} d^N x e^{-\frac{1}{2}\mathbf{x}^T(F-iG)\mathbf{x}+\mathbf{a}^T\mathbf{x}}|\mathbf{x}\rangle$ where F and G are real symmetric matrices, $\mathbf{a}$ is a vector, C is the normalization and

$$\gamma = \begin{pmatrix} F + GF^{-1}G & -GF^{-1} \\ -F^{-1}G & F^{-1} \end{pmatrix}, \ \mathbf{D} = \begin{pmatrix} GF^{-1}\mathbf{a} \\ -F^{-1}\mathbf{a} \end{pmatrix}. \tag{7.7}$$

Now, we may consider coherent superpositions $|\psi\rangle = \sum_i \alpha_i |G_i, F_i; \mathbf{a}_i\rangle$ to obtain refined approximations of a ground state. These do not possess a Gaussian characteristic function but a lengthy yet straightforward computation reveals that the corresponding characteristic function $\chi_{|\psi\rangle\langle\psi|}(\boldsymbol{\xi})$ is a sum of Gaussian functions with complex weights. Then it is immediately evident that in this description we retain the ability of efficient evaluation of all expectation values of polynomials in X and P. This allows one to establish an efficient algorithm for the approximation of ground state properties of lattice Hamiltonians that are polynomial in X and P.

7.6 Summary and Outlook

We have introduced a new variational method based on deformed weighted graph states to determine approximations to ground states of strongly interacting spin systems. The possibility to compute expectation values of local observables efficiently, together with entanglement features similar to those found in critical systems, make these states promising candidates to approximate essential features of ground states for systems with short range interactions in arbitrary geometries and spatial dimensions. One can also generalize this approach to describe the dynamics of such systems, systems with long range interactions, disordered systems, dissipative systems, systems at finite temperature and with infinite dimensional constituents. In fact, generalizations of our method that deal with these issues are possible and will be reported elsewhere.

We thank J. I. Cirac for valuable discussions and J. Eisert for suggesting the use of the Anderson bound. This work was supported by the FWF, the QIP-IRC funded by EPSRC (GR/S82176/0), the European Union (QUPRODIS, OLAQUI, SCALA, QAP), the DFG, the Leverhulme Trust (F/07 058/U), the Royal Society, and the ÖAW through project APART (W. D.). Some of the calculations have been carried using facilities of the University of Innsbruck's Konsortium Hochleistungsrechnen.

Chapter 8

PUBLICATION:
A variational method based on weighted graph states

Simon Anders, Hans J. Briegel, and Wolfgang Dür

to appear in New Journal of Physics

submitted 6 Jun 2007, accepted 10 Sep 2007, not yet published

ArXiv preprint: 0706.0423 v2 [quant-ph]

Abstract

In a recent article [Phys. Rev. Lett. **97** (2006), 107206], we have presented a class of states which is suitable as a variational set to find ground states in spin systems of arbitrary spatial dimension and with long-range entanglement. Here, we continue the exposition of our technique, extend from spin 1/2 to higher spins and use the boson Hubbard model as a non-trivial example to demonstrate our scheme.

[PACS: 02.70.c, 05.30.Jp, 03.67.Mn, 75.10.Jm, 75.40.Mg]

8.1 Introduction

Spins or harmonic oscillators on a lattice form a class of models which have been studied intensively in statistical physics. Understanding them is the key to many problems in condensed matter systems, especially regarding magnetic phenomena but also electrical and heat conduction and many other aspects. As the importance of quantum phase transitions [Sac99, Voj03] has been more and more realized, interest in the ground states of *quantum* spin models grew. While the relevance of entanglement for quantum phase transitions was initially not fully appreciated, it is now a vivid area of research (e. g., [OAFF02, VLRK03]), and many researchers feel that paying explicit attention to entanglement features is vital for further progress in numerical methods for the treatment of spin models [ON02, VPC04, Lat07]. Although quantum phase transitions nominally only occur at zero temperature, their presence has great influence on the system properties at finite temperature, namely leading to the break-down of quasiparticle descriptions. Hence, studying the ground state of spin models holds promises to understand experimentally observed features of such systems, not the least of which is high-temperature superconductivity. Finally, spin models (including bosons on a lattice) are an ideal way to model optical lattices, which are currently researched with exciting successes in theory and experiment (reviewed in [LSA$^+$07]).

While there are some exactly solvable spin models in one spatial dimension [Tak99], for nearly all models in higher dimensions approximative techniques have to be used. A variety of quite different techniques have been developed: Most prominently, these are quantum Monte Carlo techniques, where recent progress has been achieved especially in the context of the so-called world-line Monte Carlo methods ([ELM93, PST98], reviewed in [KH04]). For one-dimensional systems, extraordinary accuracy has become possible with the density matrix renormalisation group (DMRG) algorithm ([Whi92, Whi93], review [Sch05]). Recently, this algorithm was extended to allow the calculation of not only ground state properties but also of thermal states [ZV04, VGC04] and time evolutions [LXW03, MMN05, Vid04, DKSV04]. Also, an extension to higher spatial dimensions has been proposed [VC04a]. Its usability in practice has been demonstrated only very recently [MVC07].

All these variations of DMRG are based on the same class of variational[1] states, namely matrix product states [RÖ97]. We have recently found [APD$^+$06] that another class of states, namely the so-called weighted graph states (WGS), first studied in different context in [DHH$^+$05, HCDB05], is also quite promising as ansatz for variational approximation of ground states of spin systems. Its particular advantage is the unlimited amount of entanglement that can be present. Hence, we consider our technique as especially promising for systems with long-range entanglement such as critical systems. A further key difference of our states to matrix product states is that their mathematical structure does not reflect any spatial geometry (while the product of matrices in a matrix product state reflects a chain or ring geometry, as studied especially in [RÖ97, VPC04]) and hence may be expected to be equally suitable for higher dimensions (2D or 3D) as for

---

[1] Strictly speaking, only the fixed-length phase of DMRG can be called a variational method.

1D. Hence, even though we probably cannot compete with the astounding accuracy of DMRG in 1D, we aim to provide a complementary alternative to the higher-dimension generalisations of DMRG [VC04a]. In [APD+06], we presented this technique and demonstrated its use for simple spin-1/2 systems in one and two dimensions. In the present article we explain our method in much more detail, show new results we have obtained since then (especially regarding the treatment of spins higher than spin-1/2, and concerning heuristics to perform the minimizations) and tests its usefulness on practical examples. The article is self-contained and does not assume the reader's familiarity with weighted graph states or the content of [APD+06].

This article is organised as follows: We start in Sec. 8.2 by reviewing some general observations about variational methods. In Sec. 8.3 we describe our class of variational states as a generalisation of weighted graph states and discuss their parametrisation. Section 8.4 explains how reduced density matrices of these states are calculated in an efficient manner in order to be able to evaluate expectation values of observables, including energy. To test our method, we show results for calculations on two different models (namely the XY model and the Bose-Hubbard model) in Sec. 8.5. In a variational method, a crucial part is finding a state within the given class that minimises the energy as well as possible. Our techniques for doing so are the topic of Sec. 8.6. We add some further notes on the details of our numerical implementation and its performance (Sec. 3.5), and finish with a conclusion and an outlook on further work (Sec. 8.7)

8.2 General considerations on variation

For a Hamiltonian H that is too large to diagonalise one can approximate the ground state using the Rayleigh-Ritz variational method. One uses a family of states $|\Psi(\mathbf{x})\rangle$ which depend on some parameter $\mathbf{x}$. It may be better to see this as a map from a parameter space $\mathbb{R}^K$ to a Hilbert space $\mathcal{H}$:

$$\Psi : \mathbb{R}^K \to \mathcal{H}, \quad \mathbf{x} \mapsto |\Psi(\mathbf{x})\rangle \, .$$

One then solves the minimisation problem

$$E_{\text{min}} = \min_{\mathbf{x}\in\mathbb{R}^K} E(\mathbf{x}); \quad \text{with } E(\mathbf{x}) = \frac{\langle \Psi(\mathbf{x}) \,|\, H \,|\, \Psi(\mathbf{x}) \rangle}{\langle \Psi(\mathbf{x}) \,|\, \Psi(\mathbf{x}) \rangle} \tag{8.1}$$

in order to obtain an upper bound E_{min} to the ground state energy and an approximation $|\Psi(\mathbf{x}_{\text{min}})\rangle$ for the ground state.

For this to give good results, the map Ψ has to fulfil the following conditions:

(i) There must be an efficient algorithm to calculate the expectation value of observables for any state $\Psi(\mathbf{x})$. In principle, it is sufficient to be able to calculate $\langle \Psi(\mathbf{x}) \,|\, H \,|\, \Psi(\mathbf{x}) \rangle$ and $\langle \Psi(\mathbf{x}) \,|\, \Psi(\mathbf{x}) \rangle$, but if one wants not only to bound the ground state energy, but also analyse the ground state approximant $|\Psi(\mathbf{x}_{\min})\rangle$, it is desirable to be able to calculate expectation values for other observables, too.

"Efficient" means here a computation time at most polynomial in the number of parameters K. As the dimension of the Hilbert space $\mathcal{H}$ typically scales exponentially in the number N of constituents of the system, we want K to *not* do the same. Thus, the map Ψ, considered as a a family Ψ_N of maps for different system sizes N, should be such that the dimension K of its domain scales only polynomially with N, and thus logarithmically with $\dim \mathcal{H}$.

(ii) There should be reason to expect that there are states $|\Psi(\mathbf{x})\rangle$ within the range of the map Ψ that have large overlap with the true ground state or at least an energy near to the true ground state energy. As the range of Ψ is a sub-manifold of $\mathcal{H}$ of dimension at most $K \ll \dim \mathcal{H}$, this requires it either to be folded and twisted in a quite peculiar way to reach many different regions of $\mathcal{H}$, or to happen to occupy the same small part of $\mathcal{H}$ as the ground state. Typically, it is not possible to prove such a statement, and one hence has to do with heuristic arguments or numerical evidence.

(iii) There should be reason to expect that the minimisation programme (8.1) succeeds in finding a good minimum and does not get stuck in a bad local minimum. It is often not justified to hope to find the global minimum, but a local minimum of an energy only slightly higher than that of the global minimum is hardly worse.

Whether the minimisation can succeed depends on the "energy landscape", i. e. the graph of $E(\mathbf{x})$. If this landscape has many local minima, a naïve multi-start optimisation cannot succeed. Often, the number of local minima increases exponentially with N or K, which may render a method that is efficient for small systems useless for larger ones. Hence, one usually has to succeed in tailoring a heuristics that helps to find good minima for the specific kind of energy landscape one has to deal with.

One of the best studied variational methods is finite-length DMRG, and we shall illustrate the conditions given above by briefly discussing how DMRG (in the formulation of Ref. [VPC04]) fulfils them. For DMRG, the class of variational states are the matrix product states [ÖR95, RÖ97]. For an N-site matrix product state, an efficient algorithm exists to evaluate the expectation value of any observable that can be written as a sum of tensor products of local operators in time linear in N. This meets condition (i). The expectation that a matrix product state is a good approximant for the ground state of a generic 1D system (condition (ii)) is the very rationale that led White and Noack to their idea of keeping the lowest-lying eigenstates not of the short-range Hamiltonian but of the corresponding density matrix as explained e. g. in [Whi98]. The fact that condition (iii) is fulfilled, i. e. that the "sweeping procedure" of finite-length DMRG does not get stuck in local minima is somewhat mysterious, especially in the light of the possibility of construction of Hamiltonian for which this cannot be avoided [Eis06].

Nevertheless, the construction principle, as exposed in [VPC04], shows that the matrix for each site has direct influence only on this site and its neighbours, i. e. matrix product states allow for an essentially local description of states despite the existence of significant amount of entanglement. Hence, is seems natural that —barring "pathological" cases such as those discussed in [Eis06]— the local variation of matrices during sweeps allows for a good minimisation, provided the initial N-site state was chosen well (which is the task of the so-called "warm-up", which uses infinite-length DMRG). Furthermore, Wolf et al. have recently shown a close connection between approximability and Rényi entropy for matrix product states [SWVC07a].

We shall come back to some of these points when comparing our variational states with matrix product states at the end of Sec. 8.3.

8.3 The class of variational states

8.3.1 Basic idea

Our class of quantum states derives from the so-called weighted graph states, which were introduced in [DHH$^+$05, RBB03] and also used in [HCDB05]. They are a generalisation of graph states (introduced in [BR01], see [HDE$^+$06] for a review). For a Hilbert space of N qubits, they are defined as[2]

$$|\Gamma\rangle = \prod_{a=1}^{N}\prod_{b=a+1}^{N} W_{\varphi_{ab}}^{(ab)} \, |+\rangle^{\otimes N}, \tag{8.2}$$

where a product of *phase gates* W_φ is applied onto a tensor product of $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ states. These phase gates are two-qubit operation, diagonal in the computational basis, and of the form [3]

$$W_\varphi = \operatorname{diag}(1, 1, 1, e^{i\varphi}) = \exp\left[i\frac{\varphi}{2}\,(\mathbb{1} - \sigma_z) \otimes (\mathbb{1} - \sigma_z)\right]. \tag{8.3}$$

It may help to see the effect of W on small states. This is, e g., a three-qubit weighted graph state (where the qubits are numbered 1, 2, 3 from left to right):

$$W_{\varphi_{12}}^{(12)} W_{\varphi_{23}}^{(23)} W_{\varphi_{13}}^{(13)} \, |+++\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |100\rangle + |010\rangle + e^{i\varphi_{12}}\,|110\rangle + \\ + |001\rangle + e^{i\varphi_{13}}\,|101\rangle + e^{i\varphi_{23}}\,|011\rangle + e^{i(\varphi_{12}+\varphi_{13}+\varphi_{23})}\,|111\rangle \big)$$

---

[2] In this article, superscripts in parentheses always indicate the spins an operator acts on. Hence W_φ is an operator defined on a 2-spin space, while $W_\varphi^{(ab)}$ is defined on the full N-spin Hilbert space, but has support only on spins a and b.

[3] In [DHH$^+$05, HCDB05], the notation $U_{\varphi_{ab}}$ is used instead of $W_{\varphi_{ab}}^{(ab)}$. Here, we use the W to emphasise that it is a specific, and not some general unitary. Note also the absence of a minus sign in the exponential $e^{i\varphi}$, which differs from the convention used in [DHH$^+$05].

For *every* pair a, b of spins, there is a phase gate with a phase $\varphi_{ab} = \varphi_{ba}$. The key observation and starting point of the work in [DHH+05] is that even for very large N, we can efficiently calculate any reduced density matrix for a subset $A \subset \{1, 2, \dots, N\}$ of the qubits as long as the number of qubits in A (i. e. the number of qubits not traced over) is low. This calculation is efficient in the sense that the time requirement scales only polynomially in N (though exponentially in $|A|$). This is remarkable because in the generic case, the time to calculate a reduced density matrix is exponential in N, and most classes of states which allow for calculation of reduced density matrices in polynomial times are bounded in the amount of entanglement that they can contain. Especially in the case of matrix product states, this fact is the dominant reason why DMRG cannot be applied successfully for certain settings [VPC04]. Weighted graph states, on the other hand, are not bounded in the amount of their entanglement, as shall be explained in Sec. 8.3.3.

There is no guarantee that these states spread through those parts of the Hilbert space which are of interest for us, and hence, we add as many further degrees of freedom to the form (8.2) as possible without losing the ability to efficiently calculate reduced density matrices. As will be demonstrated in Sec. 8.4, the following additions do not hinder the efficiency of the reduced density matrix evaluation: (i) Let the phase gates act not simply on $|+\rangle^{\otimes N}$, but on any N-qubit product state. (ii) Even weighted superpositions of m product states can be treated, provided m is small. (iii) After the phase gates, arbitrary local unitaries may be applied.

8.3.2 Parametrisation

Deviating from the treatment in [APD+06], we develop the formulae not just for spin-1/2 particles but, more generally, for n-level systems, i. e. our states live in a Hilbert space $\mathcal{H} = (\mathbb{C}^n)^{\otimes N}$.

Superposition of product states

We start with a superposition of m product states, which we write

$$\begin{aligned}\operatorname{nrm} \sum_{j=1}^{m} \bigotimes_{a=1}^{N} \alpha_j \left(|0\rangle + d^j_{a,1} |1\rangle + d^j_{a,2} |2\rangle + \dots + d^j_{a,n-1} |n-1\rangle \right) \\ = \operatorname{nrm} \sum_{j=1}^{m} \alpha_j \bigotimes_{a=1}^{N} \sum_{s \in \mathbb{S}} d^j_{as} |s\rangle . \quad (8.4)\end{aligned}$$

The operator nrm denotes normalisation: $\operatorname{nrm} |\psi\rangle := |\psi\rangle / \| |\psi\rangle \|$. To facilitate notation, we also introduced

$V \quad := \{1, 2, \dots, N\}$ (set of spins)
$\mathbb{S} \quad := \{0, 1, \dots, n-1\}$ (set of levels)

As we normalise afterwards, we can fix the coefficient in front of $|0\rangle$ to 1: $d^j_{a,0} \equiv 1$ for all a, j. It will also be useful later to introduce the *deformation operators*

$$D_{\mathbf{d}} := \sum_{s\in\mathbb{S}} d_s \left|s\right\rangle \left\langle s\right| , \qquad \text{with } \mathbf{d} = (d_0, d_1, \ldots, d_{n-1})$$

and the n-level $|+\rangle$ state

$$|+\rangle := \frac{1}{\sqrt{n}} \sum_{s\in\mathbb{S}} |s\rangle ,$$

such that the state (8.4) can now be written in the forms

$$\operatorname{nrm}\left(\bigotimes_{s\in\mathbb{S}} D_{\mathbf{d}^j_a}\right) |+\rangle^{\otimes N} = \operatorname{nrm} \sum_{\mathbf{s}\in\mathbb{S}^N} \left(\prod_{c\in V} d^j_{c,s_c}\right) |\mathbf{s}\rangle \tag{8.5}$$

Phase gate

We entangle these product states by applying onto each pair a, b of spins a generalisation W_Φ of the 2-level phase gate W_φ from Eq. (8.3). We want to define W_Φ as general as possible, but have to meet three constraints: (i) All W_Φ have to commute (because otherwise the calculation of reduced density matrices explained later in Sec. 8.4 does not work). Hence, they have to be diagonal. (ii) W_Φ has to be unitary (for the same reason). Hence, the entries in its diagonal have to be pure phases. (iii) W_Φ should not have any parameters which can be absorbed without loss of generality into the d^j_{as}. To see, which these are, let us look at the example of $n = 3$:

$$W_\Phi \left[\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \otimes \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix} \right] = \begin{pmatrix} \Phi^{00}\,\beta_0\gamma_0 \\ \Phi^{01}\,\beta_0\gamma_1 \\ \Phi^{02}\,\beta_0\gamma_2 \\ \Phi^{10}\,\beta_1\gamma_0 \\ \Phi^{11}\,\beta_1\gamma_1 \\ \Phi^{12}\,\beta_1\gamma_2 \\ \Phi^{20}\,\beta_2\gamma_0 \\ \Phi^{21}\,\beta_2\gamma_1 \\ \Phi^{22}\,\beta_2\gamma_2 \end{pmatrix} = \begin{pmatrix} \zeta_{01} \\ \zeta_{02} \\ \zeta_{03} \\ \zeta_{10} \\ \zeta_{11} \\ \zeta_{12} \\ \zeta_{20} \\ \zeta_{21} \\ \zeta_{22} \end{pmatrix}$$

If one is given the ζ_{st} and can choose the β_s and γ_t at will, one does not need the freedom to set all entries in W_Φ. It suffices to have 4 phases:

$$W_\Phi = \operatorname{diag}\left(1, 1, 1,\ 1, e^{i\Phi^{11}}, e^{i\Phi^{12}},\ 1, e^{i\Phi^{21}}, e^{i\Phi^{22}}\right). \tag{8.6}$$

In general, for n levels, one needs to specify $(n-1)^2$ phases for each phase gate W_Φ. We denote the phases by a $(n-1) \times (n-1)$ matrix Φ_{ab} (with elements Φ^{st}_{ab}) and have

$$W_\Phi = \mathbb{1}_{n\times n} \oplus \bigoplus_{s=1}^{n-1} \operatorname{diag}\left(1, e^{i\Phi^{s1}}, e^{i\Phi^{s2}}, \ldots, e^{i\Phi^{s,n-1}}\right).$$

Defining $\Phi^{s0} \equiv 0$ and $\Phi^{0t} \equiv 0$ for all $s, t \in \mathbb{S}$, we can simply write

$$W_\Phi = \sum_{s,t\in\mathbb{S}} e^{i\Phi^{st}} |st\rangle \langle st| \tag{8.7}$$

Our variational states now take the following form:

$$|\Psi(\mathbf{x})\rangle = \mathrm{nrm} \left(\bigotimes_{a=1}^{N} U_a \right) \sum_{j=1}^{m} \alpha_j \left(\prod_{\substack{a,b\in V\\ a<b}} W^{(ab)}_{\Phi_{ab}} \right) \bigotimes_{a=1}^{N} \sum_{s\in\mathbb{S}} d^j_{as} |s\rangle \,. \tag{8.8}$$

The vector $\mathbf{x}$ is a concatenation of all the parameters that are present in the right-hand side, i. e. the (real) parameters of $\mathbf{x}$ contain the real and imaginary parts of the complex scalars d^j_{as} and α_j, the (real) entries Φ^{st}_{ab} of the phase matrices, and the parameters describing the N local unitaries $U_a \in SU(n)$, $a = 1, \ldots, N$.

Parametrisation of the unitaries

Next, we need to choose a parametrisation of $SU(n)$ in order to describe the unitary matrices U. For this, we use an isomorphism between the set $SU(n)$ of unitary $n \times n$ matrices U and the set of Hermitean $n \times n$ matrices A because Hermitean matrices are easy to parametrise. We could use (a) the matrix exponentiation $U = \exp iA$ or (b) the *Cayley transform* (introduced 1846 by Cayley, see e. g. [Puz05])

$$U = (i\mathbb{1} + A)(i\mathbb{1} - A)^{-1}. \tag{8.9}$$

To calculate these expressions numerically, we need, for (a), a matrix diagonalisation and, for the matrix inversion in (b), an LU factorisation [TB97]. We choose the Cayley transform, not only because it is slightly faster, but especially because we will later have to evaluate the derivatives of U with respect to its parameters, and while this is very involved for (a) [NH95], it is rather trivial for (b) [PlM]. (A disadvantage seems to be on the first glance that the Cayley transform is undefined if A has -1 as eigenvalue, because then, $(i\mathbb{1} - A)$ cannot be inverted. The algorithm will not, however, converge to this case, and if it happened to hit on it, the program would abort.)

Parameter count

Let us now count the number K of real parameters needed to describe a state $|\Psi(\mathbf{x})\rangle$:

- For each phase gate, we need $(n-1)^2$ real numbers. In case of one phase matrix for each pair of spins, there are $N(N-1)/2$ gates.
- For the deformations, i. e., the specification of the initial product states, we need $2mN(n-1)$ real numbers.

- For the superposition coefficients, $2m$ reals.
- An $n \times n$ Hermitean matrix is specified by $n(n-1)/2$ complex entries in one of the triangles above or below the diagonal and n real entries in the diagonal. Hence, we need for the N unitaries a total of $Nn(n+1)/2$ real parameters.

Thus, the number of parameters is

$$\begin{aligned} K &= (n-1)^2 \frac{N(N-1)}{2} + 2mN(n-1) + 2m + N\frac{n(n+1)}{2} \\ &= O(N^2 n^2 + Nnm). \end{aligned} \tag{8.10}$$

8.3.3 Entanglement properties

As already mentioned an important motivation for this work was the goal to find a class of states which exhibit strong entanglement over arbitrary distances that is somewhat "generic". After all, the limited ability to describe such entanglement is a common shortcoming of many approximation methods for many-body quantum mechanics. For the case of DMRG, this has been studied in detail in Ref. [VPC04]. There, it was shown that the matrix product states that arise during DMRG can be understood as "projections" from an auxiliary linear quantum system of the valence bond solid type [VC04b]. Hence, whenever one cuts the matrix product states "chain" into two parts, the blockwise entanglement (i. e., the entropy of the reduced density matrix of one of either part) is bounded by $2\log_2 D$, where D is the dimension of the auxiliary spins, which is equal to the number of "kept states" in DMRG parlance or the matrix size in the matrix product state picture. This explains why DMRG performs not too well when applied to long 1D systems with long-range entanglement or, more precisely, to systems where the blockwise entanglement grows with the block size.

A scaling of the entanglement is hardly avoidable when treating systems with more than one dimension. According to the various "area law" theorems and conjectures, for most systems the entanglement of a block versus the rest of the system scales linearly with the area of the interface between this block and the rest [AEPW02, PEDC05, Wol06, CEPD06, WVHC07]. Hence, for, say, a 2D system, the entanglement scales linearily with the surface area of the block and matrix product states are unable to render this feature without their matrix size growing quite fast. There are ways of replacing the matrices with higher-rank tensors to keep up with the area law, yielding so-called projected entangled pair states (PEPSs) [VC04a] but the formalism of these is rather tedious and grows more complicated with increasing spatial dimension. Also, PEPSs cannot go beyond the area law and are hence still unable to treat systems that do not follow the area law, i.e., show entanglement that scales superlinearly with the block surface, which typically is the case in critical and certain disordered systems [Kor04, KM05, CEP07, VWPC06, EO06, BCS06].

We hope that our variational method turns out to be a viable complementary method especially to this "PEPS" generalization of DMRG. To see how this claim may be substantiated, note that in the description of our states, the geometry of the system has not entered yet. Every spin is connected to every other spin by a phase gate, and we can thus model any geometry, i. e., any scheme of neighboring relations. The entanglement of a block of M spins w. r. t. the rest of the system (with $N - M$ spins) can scale with the number of spins M, i.e. with the *volume* and not with the surface area of the block [CHDB05]. Thus, the blockwise entanglement can reach the maximum value that is possible in the given Hilbert space. Other entanglement measures such as localizable entanglement between pairs of spins and also two-point correlation functions can reach their maximum value (independent of the distance), but can also show exponential or polynomial decay [DHH$^+$05]. This is already evident from the fact that 2D cluster states are within our variational class, and they reach maximum entanglement in several senses [NMDB06], e. g. the localizable entanglement between all pairs of spins is one.

8.3.4 Making use of symmetries

Symmetrising the phases

The quadratic scaling of K with the number N of spins (lattice sites) in Eq. (8.10) can be reduced to a linear scaling in case of a system Hamiltonian with translational symmetry. This is because in this case it is reasonable to assume that we do not lose precision if we let the phase matrices depend not on the absolute positions of the spins a and b but only on the position of b relative to a. More precisely, we introduce a mapping $\nu : V \times V \to \{1, \dots, R\}$, that gives the *phase index* for the spin pair a, b: the phase gate that is applied on the pair (a, b) shall be the phase matrix with number $\nu(a, b)$, and R is the total number of phase matrices. The 4th-order tensor $\Phi_{ab}^{r_1 r_2}$ now becomes a 3rd-order tensor $\Phi_{\nu(a,b)}^{r_1 r_2}$.

The mapping ν has to be constructed such that two pairs of spins, (a, b) and (c, e), get the same index, $\nu(a, b) = \nu(c, e)$, if and only if the pair (a, b) can be mapped onto (c, e) by a symmetry transformation that leaves the system Hamiltonian invariant. For the common case of a Hamiltonian that is a sum of identical terms which each act on one bond (i. e., connection of lattice sites), this is the symmetry group of the lattice. In the case of a square lattice with $N = L \times L$ sites on periodic boundary conditions (PBC), only[4]

$$R = \frac{1}{2} \left\lfloor \frac{L}{2} \right\rfloor \left(\left\lfloor \frac{L}{2} \right\rfloor - 1 \right) = O(N)$$

phase matrices are needed as can be seen from Fig. 8.1, and thus, we need only $K = O(N)$ parameters.

---

[4] The brackets $\lfloor \cdot \rfloor$ denote the floor function.

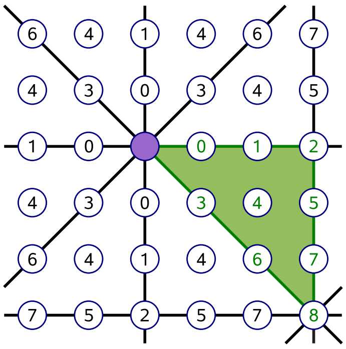


Figure 8.1: For a system on an $L \times L$ square lattice with periodic boundary conditions and a system Hamiltonian that is invariant under the lattice's symmetry group, only $R = O(N)$ phase matrices are needed. The numbers indicate the numbering of these matrices with phase indices $\nu = 0, \ldots, R-1$. The circles denote sites of a 6×6 lattice. In constructing a variational state (8.8) on this lattice, phase gates W_{Φ_ν} are performed on any pair of sites, where the phase gate acting on the purple shaded site and a site marked with the number ν uses the phase matrix Φ_ν. Translation of these markings show the phase indices for other site pairs. Note, how due to the rotation and reflection symmetries of the square lattice, the pattern of phase indices is repeated eight times.

Note also, that ν is naturally symmetric, $\nu(a,b) = \nu(b,a)$, and that this has to be reflected by a like symmetry of Φ w. r. t. its *upper* indices: $\Phi_\nu^{r_1 r_2} = \Phi_\nu^{r_2 r_1}$, which must be imposed explicitly.

Full symmetrisation

For a symmetric Hamiltonian, it seems natural to reflect this symmetry not only in the phases Φ, but also in the local, site-dependent properties, i. e., in the local unitaries U_a and the deformation parameters $\mathbf{d}_a^j$. In case of full translation symmetry, one may want to completely drop the dependence of these on the site index a. This does indeed reduce the number K of parameters significantly, but not as dramatically as in the case of phase symmetrisation. The latter reduced the scaling of K from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, while further symmetrisation of the other parameters cannot change $K = \mathcal{O}(N)$. On the other hand, the time required to calculate the energy of a given state is reduced by a factor $\mathcal{O}(N)$ in the fully symmetric case, as one needs to evaluate it for only one elementary cell of the lattice.

A good reason not to impose full symmetrisation nevertheless is the observation that for many systems, the ground state does not necessarily obey the full symmetry of the Hamiltonian due to spontaneous symmetry breaking. Even though in such a case, the ground state must be degenerate, and at least one state within the ground subspace must obey the full symmetry, this state is unlikely to be the state that is easiest to approximate within the chosen class of variational states. To give an example: The ground state of the antiferromagnetic Ising chain without transverse field is $\alpha\,|0101\ldots\rangle + \beta\,|1010\ldots\rangle$, for any α, β with $|\alpha|^2 + |\beta|^2 = 1$. Only for $\alpha = \beta$ the state is invariant under a translation of one site. However, the state most easily approximated is $\alpha = 1, \beta = 0$ (or vice versa), as it is a product state, while any other state contains long-range entanglement. If we imposed full translational symmetry onto the states, our algorithm would likely fail to find a good state. However, the example suggests a compromise between flexibility and low number of parameters: We make the local properties U_a and $\mathbf{d}_a^j$ periodic in a way that matches the expected periodicity of the spontaneously-broken ground state, e. g., in the case of the Ising chain, we may use one common unitary and one common deformation vector for all odd sites, and another unitary and another deformation vector for all even sites. However, our numerical experiments showed that this does not work particularly well: the enforcement of such symmetries introduces very many additional local minima which trap the minimization routine much too soon. The intuitive reason for this is that enforcing the symmetries amount to a cut through the energy landscape of the parameter space which seems to divide meandering troughs into separated basins.

Let us nevertheless mention two more possibilities to even further reduce the parameter scaling. (i) We can make the phase index mapping ν such that it does not depend on the geometric relation as in Fig. 8.1 but just on the scalar number of lattice steps that separates the spins, the number of phase indices scales linearly only with the length L, not with the number of sites $N = \mathcal{O}(L^{\mathcal{D}})$ (where $\mathcal{D}$ is the dimension of the system). Together with a full or periodic symmetrization of the local properties, we reach a scaling of the number of parameters $K = \mathcal{O}(L)$, which allows for a quick treatment even of 3D systems of moderate size. The accuracy achieved this way is, however, very modest.

(ii) Often, one may expect long-range entanglement to be suppressed exponentially. Then one can choose a distance threshold and fix to zero all phases between spins with a distance above this threshold. The threshold will typically be chosen of the order of the entanglement length, and as the latter usually does not increase strongly with the system size (except at criticality) one can save considerably on the number of parameters.

8.4 Evaluating observables

In order to evaluate an observable $\mathcal{O}$ with support on $A \subset V$, we need to evaluate

$$\langle \mathcal{O} \rangle_{\mathbf{x}} = \operatorname{tr} \mathcal{O} \rho_A$$

with

$$\rho_A := \operatorname{tr}_{V\setminus A} |\Psi(\mathbf{x})\rangle \langle\Psi(\mathbf{x})| \,.$$

As we shall see now, ρ_A can be calculated in time polynomial in the number $N - |A|$ of spins, the number n of levels per spin and the number m of superpositions, but exponential in the number $|A|$ of spins not traced over. Hence, the expectation value $\langle\mathcal{O}\rangle_{\mathbf{x}}$ of observables can be calculated efficiently as long as $\mathcal{O}$ is a sum of terms with small support.

In particular, we need this algorithm to evaluate the energy $\langle\Psi(\mathbf{x})| H |\Psi(\mathbf{x})\rangle$, as this is the quantity we wish to minimise. Thus, due to the scaling properties just mentioned, we require that the system Hamiltonian can be written as sum of terms with small support (as it is the case nearly always).

8.4.1 A pair of spins

To keep notation simple, we only derive the procedure to obtain the two-spin density matrix ($A = \{a, b\}$)

$$\rho_{ab} := \rho_{\{a,b\}} = \operatorname{tr}_{V\setminus\{a,b\}} |\Psi(\mathbf{x})\rangle \langle\Psi(\mathbf{x})| \,. \tag{8.11}$$

This is a generalisation of the work done in [DHH$^+$05] for spin-1/2. A further generalisation to more than two spins is easy and its result will be given at the end of this section.

The spins that we do not trace over are denoted a and b. We start by inserting Eq. (8.8) into Eq. (8.11) and pull as much as possible out of the partial trace:

$$\begin{aligned}\rho_{ab} = \operatorname{nrm} (U_a \otimes U_b)\, W_{\Phi_{ab}} \times&\\ \times \left(\sum_{j,k=1}^{m} \alpha_j \alpha_k^* \left(D_{\mathbf{d}_a^j} \otimes D_{\mathbf{d}_b^j}\right) \rho_{ab}^{jk} \left(D_{\mathbf{d}_a^k} \otimes D_{\mathbf{d}_b^k}\right)^\dagger \right) \times&\\ \times W_{\Phi_{ab}}^\dagger (U_a \otimes U_b)^\dagger \,.&\end{aligned} \tag{8.12}$$

Here, the operator nrm again means normalisation, now defined as $\operatorname{nrm}\rho := \rho/\operatorname{tr}\rho$, and the inner term ρ_{ab}^{jk} contains anything that cannot be pulled out of the partial trace:

$$\rho_{ab}^{jk} = \operatorname{tr}_{V\setminus\{a,b\}} \left|\psi_{ab}^j\right\rangle \left\langle\psi_{ab}^k\right| \tag{8.13}$$

with

$$\left|\psi_{ab}^j\right\rangle = \left(\prod_{c\in V\setminus\{a,b\}} W_{\Phi_{ac}}^{(ac)} W_{\Phi_{bc}}^{(bc)} \right) \bigotimes_{c\in V\setminus\{a,b\}} \sum_{s\in\mathbb{S}} d_{c,s}^j |s\rangle \,,$$

which is, due to Eqs. (8.5) and (8.7),

$$\left|\psi_{ab}^j\right\rangle = \sum_{\mathbf{s}\in\mathbb{S}^N} \left(\prod_{c\in V\setminus\{a,b\}} e^{i\Phi_{ac}^{s_a s_c}} e^{i\Phi_{bc}^{s_b s_c}} d_{s,s_c}^j \right) |\mathbf{s}\rangle \,.$$

Note that in the trace (8.13) all the phase gates $W_{\Phi_{ce}}$ with $c, e \notin \{a, b\}$ cancel with their Hermitean conjugate, as do all the local unitaries U_c, $c \notin \{a, b\}$. Hence, ρ_{ab} depends only on a subset of the parameters.

In order to take the trace in Eq. (8.13), we have to sum over all states $\left|\underline{\mathbf{q}}\right\rangle$, $\underline{\mathbf{q}} \in \mathbb{S}^{N-2}$, where the underline denotes that the components of $\underline{\mathbf{q}}$ are *not* indexed $(s_1, s_2, \ldots, s_{N-2})$ but rather using the elements of $V \backslash \{a, b\}$ as indices. We get

$$\begin{aligned}
\rho_{ab}^{jk} &= \sum_{\underline{\mathbf{q}} \in \mathbb{S}^{N-2}} \left\langle \underline{\mathbf{q}} \middle| \psi_{ab}^{j} \right\rangle \left\langle \psi_{ab}^{k} \middle| \underline{\mathbf{q}} \right\rangle \\
&= \sum_{\underline{\mathbf{q}} \in \mathbb{S}^{N-2}} \sum_{\mathbf{s}, \mathbf{s}' \in \mathbb{S}^{N}} \left\langle \underline{\mathbf{q}} \middle| \mathbf{s} \right\rangle \left\langle \mathbf{s}' \middle| \underline{\mathbf{q}} \right\rangle \times \\
&\qquad \times \prod_{c \in V \backslash \{a,b\}} d_{cs_c}^{j} {d_{cs_c}^{k}}^{*} \exp\left[i \left(\Phi_{ac}^{s_a s_c} + \Phi_{bc}^{s_b s_c} - \Phi_{ac}^{s'_a s'_c} - \Phi_{bc}^{s'_b s'_c} \right) \right] \\
&= \sum_{\mathbf{r}, \mathbf{r}' \in \mathbb{S}^2} \left| \mathbf{r} \right\rangle \left\langle \mathbf{r}' \right| \times \\
&\qquad \times \sum_{\underline{\mathbf{q}} \in \mathbb{S}^{N-2}} \prod_{c \in V \backslash \{a,b\}} d_{cs_c}^{j} {d_{cs_c}^{k}}^{*} \exp\left[i \left(\Phi_{ac}^{r_1 q_c} + \Phi_{bc}^{r_2 q_c} - \Phi_{ac}^{r'_1 q_c} - \Phi_{bc}^{r'_2 q_c} \right) \right]
\end{aligned}$$

In the last line, we can exchange sum and product in the following manner without changing the expression:

$$\sum_{\underline{\mathbf{q}} \in \mathbb{S}^{N-2}} \prod_{c \in V \backslash \{a,b\}} \longrightarrow \prod_{c \in V \backslash \{a,b\}} \sum_{q_c \in \mathbb{S}} .$$

This gives

$$\rho_{ab}^{jk} = \sum_{\mathbf{r}, \mathbf{r}' \in \mathbb{S}^2} \left| \mathbf{r} \right\rangle \left\langle \mathbf{r}' \right| \prod_{c \in V \backslash \{a,b\}} \underbrace{\sum_{q \in \mathbb{S}} d_{cq}^{j} d_{cq}^{k*} \exp\left[i \left(\Phi_{ac}^{r_1 q} + \Phi_{bc}^{r_2 q} - \Phi_{ac}^{r'_1 q} - \Phi_{bc}^{r'_2 q} \right) \right]}_{:= \left(\rho_{ab,c}^{jk} \right)_{\mathbf{r}, \mathbf{r}'}} \tag{8.14}$$

The sum over $\mathbb{S}$ has n terms, and $N-2$ such sums are multiplied. Hence, in order to calculate one matrix element of ρ_{ab}^{jk} we have to evaluate the underbraced term $(N-2)n$ times. This is the origin of the promised polynomial scaling for the calculation of expectation values.

Recall that ρ_{ab}^{jk} is an $n^2 \times n^2$ matrix. We will make this more explicit by writing the product as Hadamard product. The Hadamard product, denoted $\odot$, is defined as the component-wise multiplication of matrices, $(B \odot C)_{ij} := B_{ij} C_{ij}$. Its identity, denoted $\mathbb{1}_{\odot}$, is the matrix having 1 as all of its elements. Using this, we can rewrite the previous equation in a very compact form:

$$\rho_{ab}^{jk} = \bigodot_{c \in V \backslash \{a,b\}} \rho_{ab,c}^{jk} \tag{8.15}$$

where the matrix elements of $\rho^{jk}_{ab,c}$ are given by the underbraced term in Eq. (8.14). Each factor of the Hadamard product can be understood as resulting from the interaction of the spins a and b with one spin c from $V\setminus\{a,b\}$. These factors can be calculated separately because the interaction between two different spins in $V\setminus\{a,b\}$ may be and is ignored due to the cancellation of all phase gates within $V\setminus\{a,b\}$. (Cf. the remark after Eq. (8.13).)

To make this more concrete, let us look at the simple case of $n = 2$. Then (Recall that $d^j_{a,0} \equiv 1$, and $\Phi^{s,t} = 0$ for $s = 0$ or $t = 0$.)

$$\rho^{jk}_{ab} = \bigodot_{c\in V\setminus\{a,b\}} \left(\mathbb{1}_\odot + d^j_c d^{k*}_c \tilde{\rho}_{ab,c}\right)$$

with

$$\tilde{\rho}_{ab,c} = \begin{pmatrix} 1 & e^{-i\Phi^{1,1}_{ac}} & e^{-i\Phi^{1,1}_{bc}} & e^{-i(\Phi^{1,1}_{ac}+\Phi^{1,1}_{bc})} \\ e^{i\Phi^{1,1}_{ac}} & 1 & e^{i(\Phi^{1,1}_{ac}-\Phi^{1,1}_{bc})} & e^{-i\Phi^{1,1}_{bc}} \\ e^{i\Phi^{1,1}_{bc}} & e^{i(-\Phi^{1,1}_{ac}+\Phi^{1,1}_{bc})} & 1 & e^{-i\Phi^{1,1}_{ac}} \\ e^{i(\Phi^{1,1}_{ac}+\Phi^{1,1}_{bc})} & e^{i\Phi^{1,1}_{bc}} & e^{i\Phi^{1,1}_{ac}} & 1 \end{pmatrix}, \tag{8.16}$$

which is the formula given in [DHH$^+$05].

8.4.2 Several spins

For reference, we give the result for density matrices for not simply two spins a, b, but arbitrary numbers of spins, given in a set A:

$$\begin{aligned} \rho_A &:= \operatorname{tr}_{V\setminus A} |\Psi(\mathbf{x})\rangle \langle\Psi(\mathbf{x})| \\ &= \operatorname{nrm} \left(\bigotimes_{a\in A} U_a\right) \left(\prod_{\substack{a,b\in A\\ a<b}} W^{(\tau(a),\tau(b))}_{\Phi_{ab}}\right) \times \\ &\quad \times \left[\sum_{j,k=1}^{m} \alpha_j \alpha_k^* \left(\bigotimes_{a\in A} D_{\mathbf{d}^j_a}\right) \left(\bigodot_{c\in V\setminus A} \rho^{jk}_{A,c}\right) \left(\bigotimes_{a\in A} D_{\mathbf{d}^k_a}\right)^\dagger\right] \times \\ &\quad \times \left(\prod_{\substack{a,b\in A\\ a<b}} W^{(\tau(a),\tau(b))}_{\Phi_{ab}}\right)^\dagger \left(\bigotimes_{a\in A} U_a\right)^\dagger \end{aligned} \tag{8.17}$$

with

$$\rho^{jk}_{A,c} = \left(\sum_{s\in\mathbb{S}} d^j_{c,s} d^{k\,*}_{c,s} \exp\left[i \sum_{a\in A} \left(\Phi^{r_{\tau(a)},s}_{ac} - \Phi^{r'_{\tau(a)},s}_{ac}\right)\right]\right)_{\mathbf{r},\mathbf{r}'\in\mathbb{S}^{|A|}}. \tag{8.18}$$

The mapping $\tau : A \to \{1,\dots,|A|\}$ gives here the index that spin $a \in A$ gets within the density matrix ρ_A (i.e., in the 2-spin case of ρ_{ab}, $\tau(a) = 1$ and $\tau(b) = 2$.).

It is also useful to observe that

$$\left(\bigotimes_{a\in A} D_{\mathbf{d}_a^j}\right) \rho \left(\bigotimes_{a\in A} D_{\mathbf{d}_a^k}\right)^\dagger = \left|D_A^j\right\rangle \left\langle D_A^k\right| \odot \rho \tag{8.19}$$

with

$$\left|D_A^j\right\rangle = \sum_{\mathbf{r}\in\mathbf{S}^{|A|}} |\mathbf{r}\rangle \prod_{a\in A} d^j_{a,r_{\tau(a)}} \tag{8.20}$$

This formula comes from the observation that a matrix product of a diagonal matrix, an arbitrary matrix, and another diagonal matrix can be written as Hadamard product:

$$(\operatorname{diag}\mathbf{u})\ A\,(\operatorname{diag}\mathbf{v}) = A \odot \left(\mathbf{u}\mathbf{v}^\dagger\right) \tag{8.21}$$

Further, for the numerics, one may use

$$\prod_{\substack{a,b\in A\\ a<b}} W_{\Phi_{ab}}^{(\tau(a),\tau(b))} = \operatorname{diag}\left(\exp\left[i\sum_{\substack{a,b\in A\\ a<b}} \Phi_{ab}^{r_{\tau(a)},r_{\tau(b)}}\right]\right)_{\mathbf{r}\in\mathbf{S}^{|A|}}.$$

8.5 Demonstration for two models

To approximate a ground state, we have to vary the parameters in order to minimize the energy. Before we explain our techniques to achieve this we show the results of such minimizations for two different models to demonstrate the performance of our technique. The two model systems, namely the XY model and the Bose-Hubbard model, are presented in the two following subsections. For each of the two models, we have used a different implementation (see Sec. 8.6 for details) and different heuristics for the global minimisation. Hence, we shall use these results as examples when explaining these heuristics in Sec. 8.6. As the second implementation is newer and its heuristics more sophisticated, the results for the Bose-Hubbard model are more convincing. Nevertheless, we also present our results for the XY model, as the old heuristics provides illuminating insights into important aspects of our methods behaviour. The examples with the XY model are a continuation of the examples for the Ising model (which is a special case of the XY model) already given in [APD$^+$06].

8.5.1 The XY model with transverse field

The *XY model with transverse field* for a system of spin-1/2 particles on a lattice is given by the Hamiltonian

$$H = \sum_{\{a,b\}\in\mathcal{B}} \left(\frac{1+\gamma}{2}\sigma_x^{(a)}\sigma_x^{(b)} + \frac{1-\gamma}{2}\sigma_y^{(b)}\sigma_y^{(b)}\right) + \sum_{a\in V} B\sigma_z^{(a)},$$

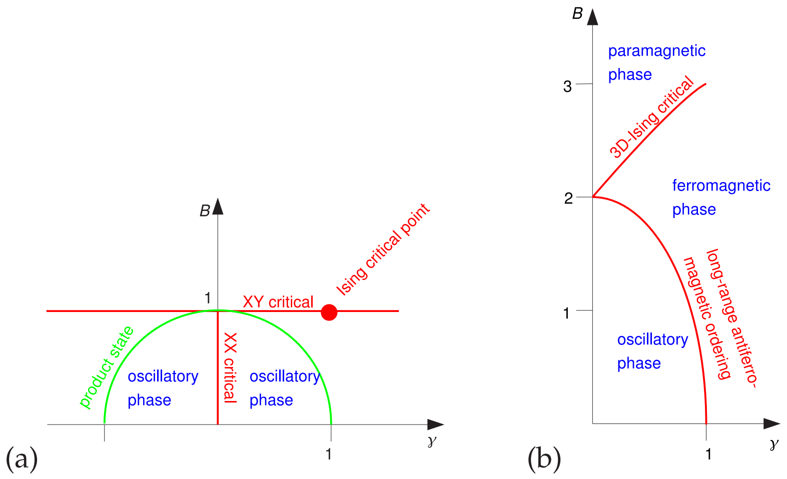


Figure 8.2: Phase diagram of the XY model, (a) for 1D (according to [BM71]), (b) for 2D (according to [Hen84]).

where $\sigma_{x,y,z}$ are the Pauli matrices, $\mathcal{B}$ is the set of nearest neighbours, B the transverse field and γ is called the asymmetry. For $\gamma = 0$, we get as special case the *XX model*, and for $\gamma = 1$, we get the *Ising model*.

One dimension (spin rings)

For 1D, the XY model with transverse field can be diagonalised using a Jordan-Wigner and then a Bogoliubov transformation (the latter is trivial for $\gamma = 1$). Correlations have been studied in early work in [Pfe69] (Ising) and [Kat62, BM71] (XY). The latter article also gives the phase diagram of the 1D XY system (reproduced in Fig. 8.2a). The entanglement properties of these phases and their transitions have recently found much interest. The behaviour first indicated by numerical studies [VLRK03, LRV04] was soon confirmed by analytic calculations [JK04, Pes04, IJK05].

Our technique seems to be suited to study this model: the results are quite precise. Fig. 8.3 shows a transition through the Ising critical point. The curves show the XX correlations for different spin-spin distances in a ring of $N = 16$ spins.

As our technique tends to spontaneously break symmetry where the true ground state does not, it makes sense to plot the two-points correlations[5] for many different values of the parameters of the Hamiltonian (here: B and γ) in order to spot phase transitions. We find that it works better to plot correlations for a specific distance than to estimate correlation lengths from the

---

[5] We either plot the XX correlations or the maximum singular value of the correlation matrix $\left(\langle\sigma_i^{(a)}\sigma_j^{(b)}\rangle - \langle\sigma_i^{(a)}\rangle\langle\sigma_j^{(b)}\rangle\right)_{i,j=x,y,z}$

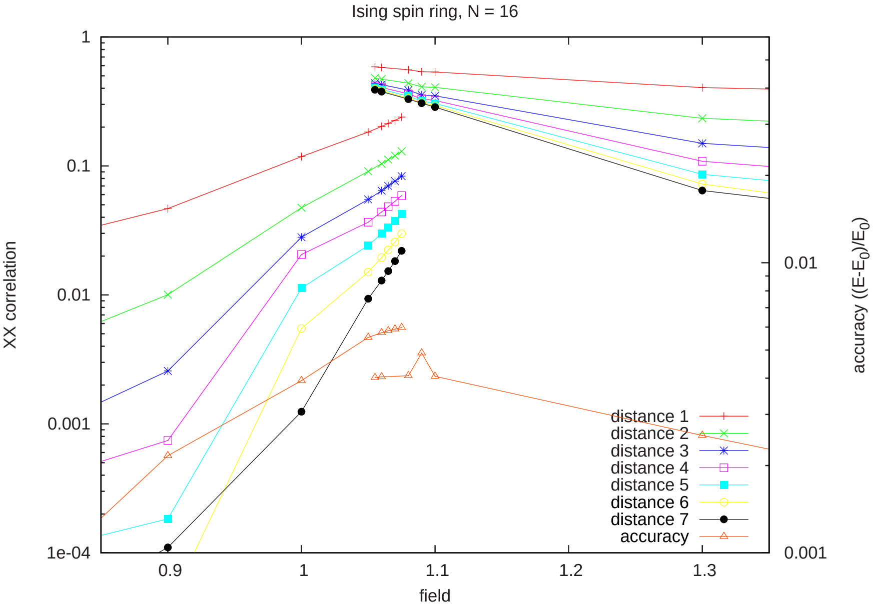


Figure 8.3: Approximation of the ground state of an Ising ring with 16 spins, calculated for $m = 4$, i. e., the phase gates act on superpositions of 4 product states: The orange triangles shows the deviation of the variational energy from the true ground state energy as obtained from the exact solution. The other symbols show the mean value of XX correlations for different spin-spin distances. As a guide to the eye, symbols for the same distance are connected by lines. It is evident that one seems to need two lines to connect the set of data points for each distance, as a single line would "jump" in zig-zag near the critical point. In other words: There seem to be two basins of attraction for the minimizer, corresponding to the $B < 1$ and $B > 1$ phase, and near the critical point, the minimizer may either fall into one basin or the other. The use of the sweeping technique, to be discussed in Sec. 8.6.3, allows to get around this undesirable behavior and direct the minimizer to minima with smaller basins of attraction which better resemble the true ground state in the area of influence of the quantum phase transition. Hence, this plot is not meant to show a good result, but rather illustrate the kind of failure that motivates and necessitates the sweeping technique.

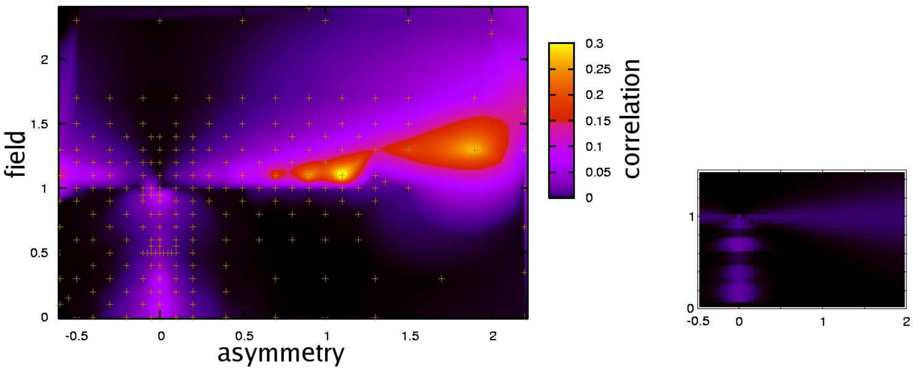


Figure 8.4: Correlations (maximum SV of correlation matrix for distance 6) of the 1D XY model, calculated for a ring of 16 spins, and with $m = 3$. The brown crosses show the positions of the data points, the colour surface in between is interpolated using Sibson's method (see Sec. 8.A.5 and be careful to not be mislead by artifact of the interpolation, such as apparent features at regions with too sparse data points). The correlations appear at that regions that also have high entanglement in the thermodynamic limit. (Compare with Fig. 3 in [LLRV05].) However, this agreement is, unfortunately, only qualitative: A comparison with the exact result for the considered finite size case, shown in the small plot to the right, reveals that correlations are over-estimated significantly.

data because the system is still so small that the exponential decay of correlations is masked by boundary effects . Fig. 8.4 shows such a plot for the 1D XY model. As is to be expected, one sees that near critical regions correlations are much stronger. (For the infinite chain, the critical regions are: XX criticality at $\gamma = 0$ for $0 < B < 1$ and XY criticality[6] for $B = 1$ [BM71].) The spread of the areas of high correlation around the critical regions of the infinite chain looks similar areas of high entropy identified in [LLRV05] – compare with Fig. 3 in that article (and note that there, entropy is small around $\gamma = 0,\ B = 1$ despite the critical nature of this point – a feature also seen in our plot of correlations.) Had we not known the critical regions, it is not merited to conclude that the system is critical where the correlations are strong, as the system size and the correlation distance is surely to small for this. We rather suggest to use a plot of this kind for a first look at a yet unstudied Hamiltonian. Regions of high correlations may suggest points in parameter space for which numerical calculations for different system sizes may give interesting results.

Another interesting feature of the 1D XY model is the Baruch-McCoy circle, which is the defined by $B^2 + \gamma^2 = 1$. On this circle, the ground state has product form [BM71]. Our approach accurately reproduces the vanishing of

---

[6] Strictly speaking the model is XY critical only for $B = 1, \delta \neq 1$, and Ising critical for $B = \delta = 1$.

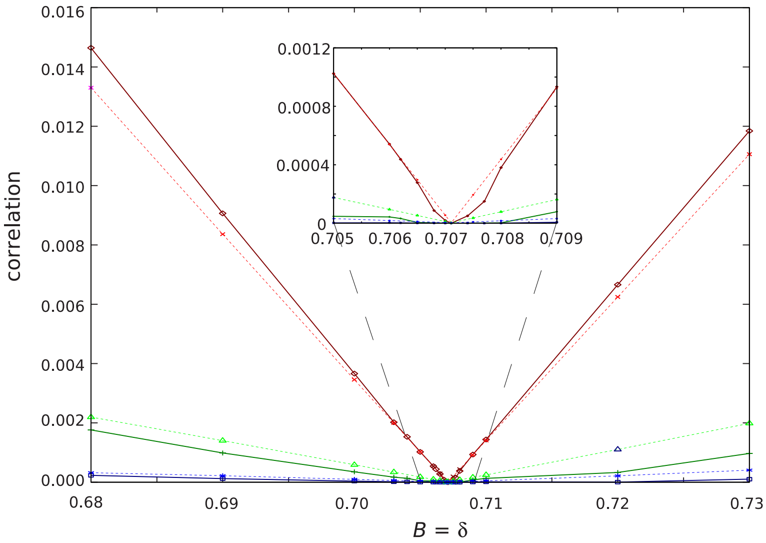


Figure 8.5: Correlations (maximum SV of correlation matrix) in the vicinity of the Baruch-McCoy circle. Along this circle, which is defined by $B^2 + \gamma^2 = 1$, the ground state of the XY model has product form. This is nicely reproduced by our numerics: At $B = \delta = 1/\sqrt{2}$, all correlations vanish. To show this, we here plot the correlations for spin-spin distances 1 (red), 2 (green), and 3 (blue) for a cut along the line $B = \delta$, i.e., radially through the circle. The x axis is the value of $B = \delta$. Calculated for a ring of 16 spins and $m = 3$ as in Fig. 8.4. The solid, dark lines are results from the variation, the dashed, light lines are exact values.

all correlations as one approaches a point on the Baruch-McCoy line (Fig. 8.5).

Two dimensions

The 2D XY model with transverse field has been studied in [Hen84]. The main result of the latter treatment is illustrated by Fig. 8.2b.

In order to demonstrate our scheme in a 2D setting, we have done calculations for a torus (i. e., a square with periodic boundary conditions) of 6×6 spins. We fixed the asymmetry at $\gamma = 0.65$ and varied the field strength B from 0 to 4.5 in order to cross both of the phase transitions indicated in Fig. 8.2b. The results, shown in Fig. 8.6, show prominent kinks at the expected positions of the phase transitions, and the correlations fall off in a roughly exponential manner with distance as expected. We still see additional jumps due to convergence into wrong basins, and this prompted us to seek a means to avoid this, namely the sweeping technique. The plots in the following section have been obtained this way and hence do not show such strong jumps.

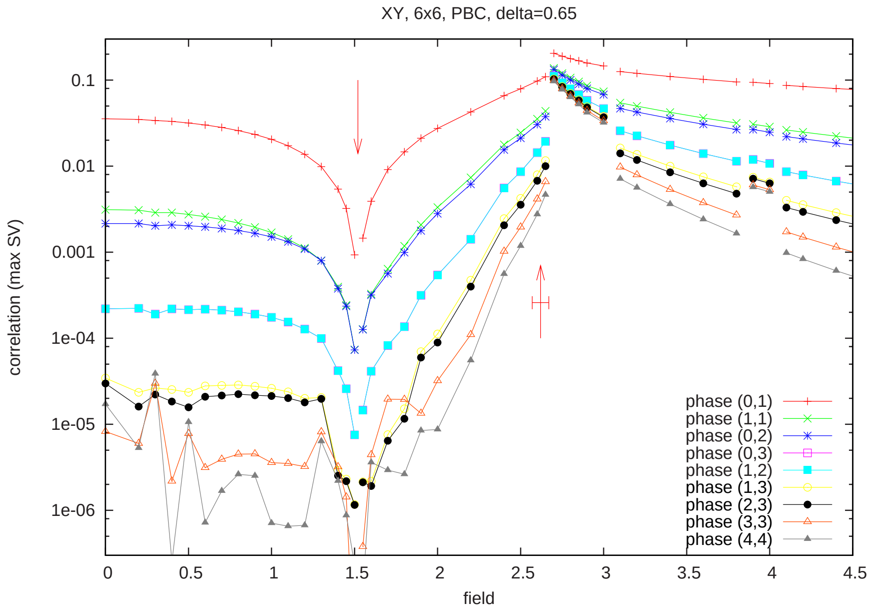


Figure 8.6: Correlations (maximum SV of correlation matrix) for the 2D XY model, calculated for a torus of 6×6 along a cross section through the phase plane of Fig. 8.2b along the line $\gamma = 0.65$ (number of superposed states: $m = 3$). The red arrows show the positions of the two phase boundaries for the *infinite* case (according to [Hen84], cf. Fig. 8.2b.) As this plot has been produced without use of the sweeping technique, some instances of convergence to the wrong minimum are evident from the jumps at $B > 3$. (Compare with the discussion at Fig. 8.3). The different curves show the correlation for spin pairs with distance (d_x, d_y) in x and y direction.

8.5.2 Bose-Hubbard model

The Bose-Hubbard model is defined for a system of harmonic oscillators, arranged in a lattice, and is described by the Hamiltonian

$$H = -J \sum_{\{a,b\}\in\mathcal{B}} \left(\hat{b}_a^\dagger \hat{b}_b + \text{H.c.}\right) + U \sum_{a\in V} \hat{n}_a \left(\hat{n}_a - 1\right)/2 - \mu \sum_{a\in V} \hat{n}_a \tag{8.22}$$

As before, V is the set of all lattice sites, and $\mathcal{B}$ the set of all unordered pairs of nearest neighbour. The operators $\hat{b}_a^\dagger$ and $\hat{b}_a$ denote the ladder operators to create and annihilate a bosonic excitation of the oscillator at site a, and $\hat{n}_a = \hat{b}_a^\dagger \hat{b}_a$ is the number operator. The first term, called the *hopping term* describes the "hopping" of an excitation from a site a to a neighbouring site b, a process which occurs with the *hopping strength* J. The second term describes the repulsion between several bosons on the same site. To fix our energy scale, we set the repulsion U to 1 in the following, i. e., all dimensionless energies are to be understood in units of U.[7] The last term is relevant if the particle number is not fixed, which it is in fact not in our case. Then, assigning a value to the chemical potential μ allows to choose the mean density of the ground state.

The Bose-Hubbard Hamiltonian is of interest due to its rich phase diagram, first exposed in [FWGF89]. While its original motivation was the description of certain structured solid state systems such as arrays of Josephson junctions, interest in the system increased significantly with the discovery that it can be realized with cold atoms in optical lattices [JBC+98] and with the spectacular experimental demonstration of this fact [GME+02], where a transition from the Mott insulator phase to the superfluid phase and back was observed. (For a review, see [LSA+07]).

In order to simulate a bosonic system with our ansatz, we restrict the number of occupations at each site. For all the following calculations, we set the dimension of each site to $n = 5$, i. e., the maximum occupation per site it $n - 1 = 4$. The creation operator $\hat{b}^\dagger$ is defined such that $\hat{b}^\dagger \left|n-1\right\rangle = 0$ in order to truncate the Hilbert space.

A good way to distinguish the Mott insulator from the superfluid phase it to look at the mean compressibility

$$\kappa = \frac{1}{N} \sum_{a\in V} \sqrt{\left(\langle \hat{n}_a^2 \rangle - \langle \hat{n}_a \rangle^2\right)},$$

which is strongly suppressed in the Mott insulator phase. Our results shows the form of the phase diagram in impressive clarity (Fig. 8.7). Although each data point only required a rather quick and rough calculation, one gets a

---

[7] When comparing with other literature, care has to be taken that many authors use the alternative convention to set $J \equiv 1$. Also, J is often denoted t.

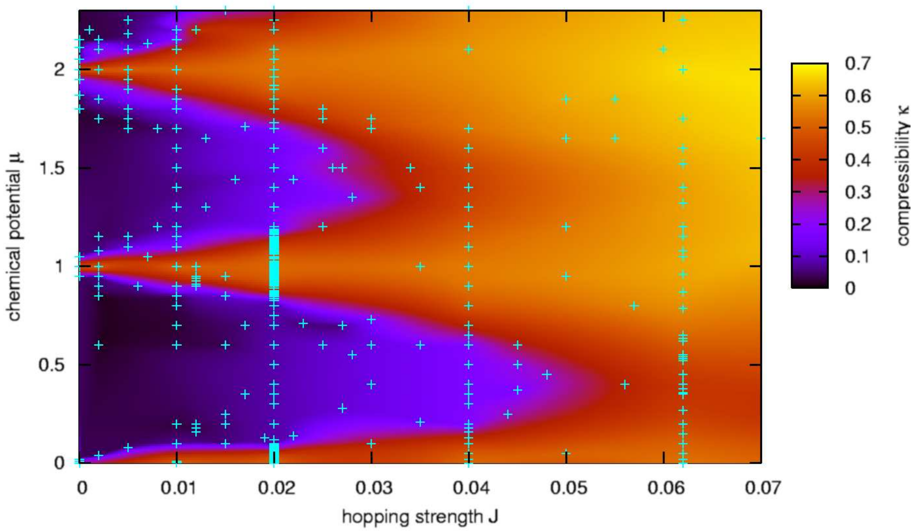


Figure 8.7: Mean compressibility κ of a 4×4 lattice (with PBC) of Bose-Hubbard sites as function of the Hamiltonian parameters J and μ. One can clearly see the Mott insulator lobes —characterised by low values (0 in the infinite case) of κ— for densities $\rho = 1, 2, 3$. and the surrounding superfluid phase. The crosses in cyan mark the parameter points for which a calculation was performed. To depict the values, an surface was interpolated between these points and is used to colour the plot. Unfortunately, we could not fully get rid of artefacts due to the interpolation (see 8.A.5), and at those regions where the density of data points varies the contour lines become distorted. The resulting "wobbling" at regions with sparse data is hence not genuine and should vanish if one adds more data points. As cuts through the plane do not suffer from these presentation problems, we have calculated much more points for three fixed values of J and show these cuts in Fig. 8.8.

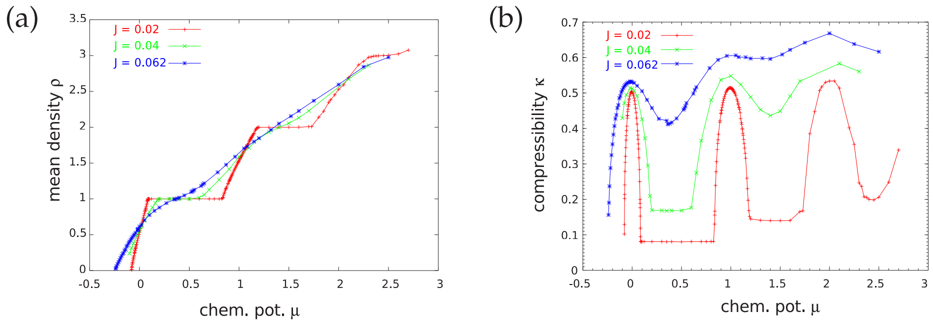


Figure 8.8: As evident from the crosses in Fig. 8.7, we have many data point for $J = 0.02, 0.04, 0.062$, which allow us to plot vertical cuts through the plane of Fig. 8.7. Here, we show the values of the average occupation number per site ρ (a) and the compressibility κ (b) for the mentioned three values of the hopping strengths J. The curve for $J = 0.02$ looks smoothest because these points have been calculated to higher accuracy (cf. Fig. 8.10).

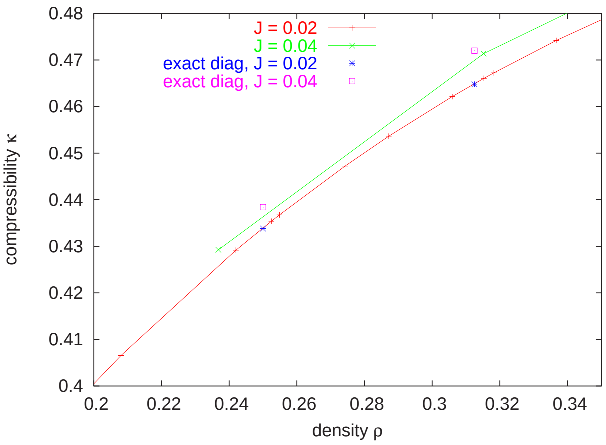


Figure 8.9: For values of the density ρ corresponding to very low (and integer) total particle numbers, we can obtain exact values for energy E and compressibility κ from diagonalisation of the full Hamiltonian. This shows that our method has good accuracy. (Note that the plot corresponds to a very small section of the steep left-most flank in Fig. 8.8b.) For the $J = 0.02$ curve, which has been calculated to especially high accuracy, the exact values for $\rho N = 4, 5$ (i. e., as $N = 4^2$: $\rho = 0.25, 0.325$) coincide with the approximated and interpolated values with an absolute deviation of only 10^{-4}. Even for the values for $J = 0.04$, which have been obtained with much fewer sweeping steps, the accuracy is below $2 \cdot 10^{-3}$.

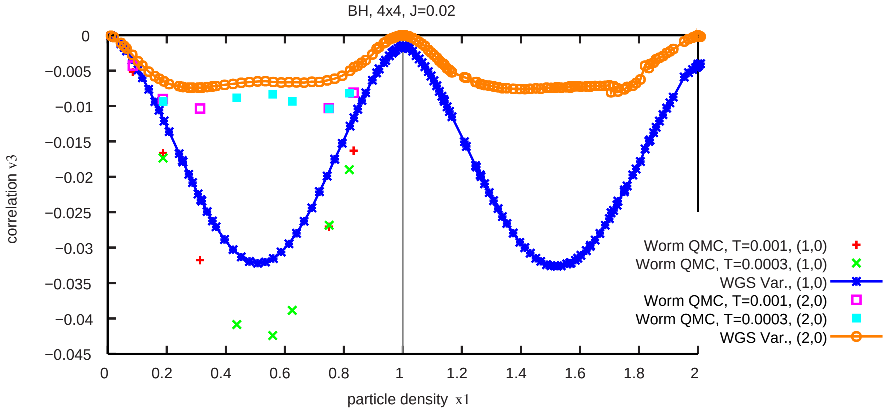


Figure 8.10: Non-local properties of the ground state are much harder to obtain than local ones. In order to see whether our technique is also capable of yielding good results here, we calculated (for $J = 0.02$ and varying values of ρ for the 4x4 sites Bose-Hubbard system) the density-density correlation $\gamma := \frac{1}{N}\sum_{a\in V}\langle \hat{n}_a \hat{n}_{a'}\rangle - \rho^2$ where a' denotes the site that is one or two lattice step(s) to the right (denoted with (1,0) and (2,0). (Correlations of this kind have, incidentally, been studied recently in [PRB06].) The connected points show the results obtained with our approximation, the isolated points have been calculated using the worm code (quantum Monte Carlo technique) [TAH98] of the ALPS project [ADG$^+$05] (at finite, but low temperature). The agreement is qualitatively OK but quantitatively not too precise. (Actually, the precision of two-point correlation is unfortunately insufficient to obtain a good picture of the momentum distribution from their Fourier transformation.)

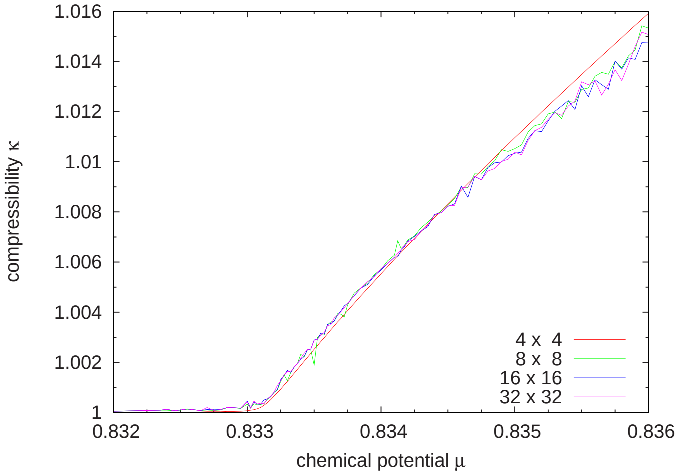


Figure 8.11: Ground state approximation for Bose-Hubbard systems with up to 32×32 sites. The figure shows the compressibility κ as function of the chemical potential μ at the phase transition between the $n = 1$ Mott lobe and the superfluid phase above of it (for a coupling of $J = 0.02U$). As one can see, the numerics cope with the large amount of sites but fails to converge with a precision sufficient to make out clear finite-size scaling trends. The reason seems to be that the number of local minima increases very strongly with system size: While the use of sweeping technique allowed to get a smooth curve for the 4×4 case, our attempts on the larger systems failed to get smoother than shown here. (We put considerably more effort in the data for $\mu < .8345$, but even there, the curves are still very noisy.)

good overview of the ground state properties in dependence of the Hamiltonian parameters J and μ. To better show the quantitative features, we have also plotted vertical cuts through the (J, μ) plane (Fig. 8.8). Especially for the points at $J = 0.02$, the zigzag sweeping technique (described later in Sec. 8.6.3) was used to improve accuracy by more than an order of magnitude. This can also be seen from Fig. 8.9: In this plot, we compare the observable κ, calculated for our approximant states, with exact values. To allow for this comparison, we have included values from exact Lanczos diagonalisation. We are grateful to G. Pupillo, who supplied these numbers to us. He used a program, written for another project and using ARPACK [LMSY96], that allows to diagonalize a small Bose-Hubbard system exactly if the number of particles is small as well. For a 4 × 4 system, up to approx. 6 particles in the 16 sites can be treated. This corresponds to the very beginning of the plots of Fig. 8.8, which we have magnified in Fig. 8.9. The accuracy of 10^{-4} for the compressibility is competing well with the precision attainable with quantum Monte Carlo techniques.

While the compressibility is a local observable, the more challenging task is to study non-local observables such as density-density correlations of the form

$$\gamma_\nu := \frac{1}{N} \sum_{a \in V} \left[\langle \hat{n}_a \hat{n}_{a'} \rangle - \left(\sum_{a \in V} \langle \hat{n}_a \rangle \right)^2 \right],$$

where a' is the site which has a fixed position relative to a, i. e. $\nu(a, a') = \nu(a)$. In Fig. 8.10, we attempt this task for a 4 × 4 lattice. Fig. 8.11 shows calculations for larger systems, up to 32 × 32 sites. While the noise present in the latter plot is small on an absolute scale (note that the plot zooms in to a quite small parameter region) is is unfortunately still too large to prevent us from doing finite-size scaling.

8.6 Performing the minimisation

Usually, the Hamiltonian of a spin system is given in the form of a sum of terms each of which has support on only a small number of spins – one or two in most physical cases. When the terms acting on single spins are absorbed into those acting on two spins, such a Hamiltonian can be written as

$$H = \sum_{(a,b) \in \mathcal{B}} H_{ab}^{(ab)},$$

where $\mathcal{B}$ is the set of all pairs of spins, on which a term acts jointly. These pairs are called *bonds* in the following, and they typically (but not necessarily) form a regular lattice. The *bond Hamiltonians* H_{ab} may all be equal or not, and only in the former case, the simplifications of Sec. 8.3.4 can be used.

The minimisation problem Eq. (8.1) that we have to solve then takes the form

$$E_{\min} = \min_{\mathbf{x}\in\mathbb{R}^K} E(\mathbf{x}); \quad \text{with } E(\mathbf{x}) = \sum_{(a,b)\in\mathcal{B}} \operatorname{tr} H_{ab}\rho_{ab}$$

Finding a minimum of a general function of many parameters is a thoroughly researched but intrinsically hard problem. Our approach is described in the following. As we do not assume the reader's familiarity with numerical optimisation, we will explain some textbook knowledge.

8.6.1 Local search

Given a starting point $\mathbf{x}_0 \in \mathbb{R}^K$ in parameter space, the problem of *local search* or *local minimisation* is the task of finding a local minimum in the vicinity of $\mathbf{x}_0$. An exhaustive treatment of this topic can be found in the standard textbook [NW99] which covers all of the algorithms mentioned in the following in detail. In our case, we have to deal with *unconstrained* (i. e., all values of the unbounded space $\mathbb{R}^K$ are admitted) *nonlinear* (i. e., the energy function $E(\mathbf{x})$ does not have any simple structure that would allow the use of a more powerful, specialised algorithm) local minimisation. Algorithms for this case come in two classes: So-called *direct methods* only require a means to evaluate the function at any given point, while *gradient-based methods* also require a means to obtain the gradient $\boldsymbol{\nabla}_{\mathbf{x}} E(\mathbf{x})$ at any given point.

Direct methods are convenient, but comparably slow. For very small systems (chains of up to 6 spin-1/2 sites, corresponding to less than 100 parameters), we could achieve convergence with direct methods, using the two most common ones, Nelder-Mead [NM64] and Powell [Pow64] minimisation, with Powell minimization converging faster.

For any meaningful system size, however, direct methods are much too slow. Hence, we coded routines to obtain the derivatives of E w. r. t. all kinds of parameters.[8] This required rather tedious calculation and coding, and the formulae and their derivation are summarised in 8.B.

Using the gradient functions, we tried the standard minimisation methods the literature offers, namely the Fletcher-Reeves conjugate-gradient method, the Polar-Ribière conjugate-gradient method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. We started using the implementations provided by the GNU Scientific Library [GTJ+03], which, however, turned out to be not robust enough. Nevertheless, it could be established that convergence speed for our problem is as usually expected, i. e. Polar-Ribiére (the oldest of the algorithms, from 1964) performs worst and BFGS (the newest, from the 1970s) performs best.

---

[8] Note that it is not helpful to obtain the gradient by numerical differentiation, as this is hardly faster than using a direct method.

BFGS is a so-called quasi-Newton or Davidon algorithm. This means, it uses the gradients obtained at the points visited so far to build up an estimate to the Hessian of the function (assuming that the Hessian varies only slowly). The approximation to the Hessian (or more precisely, to the inverse of the Hessian) is then used to make a good guess for the next step. As the approximant is, like the Hessian, a $K \times K$ matrix, updating it at each step requires $\mathcal{O}(K^2)$ steps, which scales worse than the calculation of $E(\mathbf{x})$, and hence, maintaining the BFGS data becomes more expensive than evaluating the function.

The textbook solution to this problem is to use the "limited memory" variant[9] of BFGS, which is known as L-BFGS and stores only a list of the last, say 25, gradients, and uses this data to produce a Hessian approximant "on the fly" [BLN95]. We have used the L-BFGS-B Fortran code [ZBLN97], which is very robust, not the least due to the excellent line search routine [MT94] that it uses.

A problem is the stop condition, which decides when convergence is assumed. We have tried several approaches: watching the norm of the gradient, the size of the steps, and the difference of the function value per step; these either taken point-wise, or averaged over the last, say 30, or 100, steps, or taken the maximum from the last 30-or-so steps. All this could not clearly predict convergence, as there seem to be long shallow slides, which tempt one to stop minimisation prematurely. In the end, we found that waiting until progress gets below machine precision is most viable. However, in the sweeping technique, described later, the minimization can be stopped once it seems advantageous to first continue with a neighbor.

The described technique only allows to find local minima. How do we find a good local minimum, or even the global one? Although the literature discusses many different heuristics and algorithms, it is a far from trivial task to find a good scheme. We have developed and tested two different heuristics, which shall be explained in the following two subsections. Both heuristics are two-phase methods [Sch02], i. e., they combine a global driving scheme, that chooses points to start a local search from, with the local-search algorithm just discussed.

In both cases the minimization for a specific tuple of Hamiltonian parameters is typically performed several times. Whenever a new energy value is found which is lower than all values that have been found so far for this parameter tuple, the previous energy value and corresponding state is replaced by the new one. Hence, the longer one performs these heuristics the nearer one gets to the true ground state (or more precisely: to the lowest lying state within the variational class). We emphasize that we never discard a data

---

[9] The term "limited memory" shows that the problem of keeping the full matrix was then, in the 1980s, not so much seen in the time it takes to update the matrix but rather simple in the fact that a large matrix might not fit into the memory of the computer.

point unless it has been "underbid" by a new calculation. This makes the result objective even if subjective judgement has been used in carrying out the heuristics.

8.6.2 Multi-start and step-wise adding of superpositions

For the calculation of the results for the XY model (presented in Sec. 8.5.1), we tried several different heuristics in order to "move around" local minima. We finally settled for the "multi-start" scheme described now, which turned out to work best, at least for the examples we studied: We started by choosing the parameters $\mathbf{x}_0$ for a state $|\Psi(\mathbf{x}_0)\rangle$ with $m = 1$ (i. e., without superpositions) uniformly at random from $[-5;5]^K$ and the used the L-BFGS algorithm (as explained in Sec. 8.6.1) to go downhill from there towards a minimum. We allowed this minimisation only to run for a limited number of function evaluations (typically a few hundred, or up to 1000), and then restarted with another randomly chosen initial point. Having done a number (say 15) of such "trial runs", the one that reached the lowest energy within the limited number of steps is kept, the other data is discarded. The best run is now allowed to continue for a significantly longer time, until the maximum number of "main run" steps (typically, several thousand function evaluation) is exhausted or the energy change falls below machine precision. Then, we increased the number m of superpositions by one. This makes the parameter vector $\mathbf{x}$ longer, i. e. $2N + 2$ real numbers have to be added (for N complex deformation parameters and one complex superposition coefficient, cf. Eq. (8.10) for $n = 2$). These are again chosen at random, but without changing the parameter values that have already been found. (It also helps to choose the new value for α_m with small modulus, such that the new parameters do not let the state stray to far from the already established good state.) Again, a number of trial runs is started, with different random numbers to extend the parameter vector, and the best one is allowed to continue for many more steps in the main run phase. This iterative extending of parameter values was looped until m reached a certain value. This values does not have to be very large: for the results presented in Sec. 8.5.1, $m = 3$ was sufficient.

A disadvantage of this heuristics is evident in Fig. 8.3: Some points are much worse than their neighbours. For example, the point $B = 1.09$ shows a sharp peak towards worse accuracy (orange line), while its neighbours to *both* sides are better. What happened is that near the phase transition, the two phases compete to govern the ground state, and once the minimiser gets trapped by the catch-basin of one of the two phases it cannot switch to the other one. In most cases, the multi-start scheme will allow us to enter the main run within the catch-basin of the correct phase. If, however, the minimum energy of the two competing phases are very close, they cannot be distinguished during the short and rough trial runs, and it depends on mere chance to which phase we converge. The obvious solution is to use the value

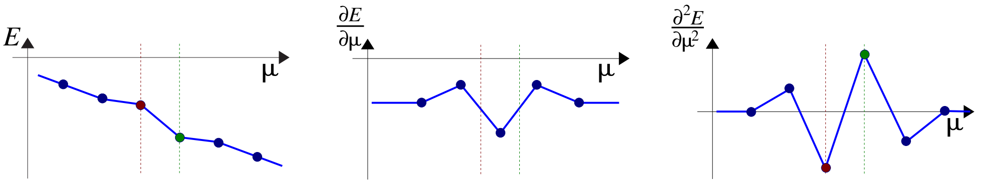


Figure 8.12: Using derivatives to locally judge the quality of the approximant. Red (left) dashed vertical line: bad point, green (right) dashed vertical line: good point. For explanation see main text (Sec. 8.6.3).

of neighbours which seem to have converged to lower energies as starting points in order to see if this allows to get to lower energies. This is the strategy that we tried next.

8.6.3 Zigzag sweeping

All the results for the Bose-Hubbard model (as presented in Sec. 8.5.2) have been obtained without the use superpositions to the right of the phase gates, i. e., with $m = 1$. Accuracy was instead improved in an iterative way using the following heuristics: Start minimisations from parameter vectors chosen at random for a variety of different values of Hamiltonian parameters (i. e., chemical potential μ and on-site repulsion J, for the Bose-Hubbard Hamiltonian) within the area of interest. Once the minimisations have converged more or less, compare each point with its neighbours. If one point looks better than a neighbouring, use this point's parameter vector to start a minimisation for the neighbouring point's Hamiltonian parameters.

In order to see how to do this in an objective way, look at the example of the red curve of Fig. 8.8. There, $J = 0.02$ is kept fixed and μ varies from -0.08 to 2.7. The data points are spaced rather closely (μ varies in steps of 0.003 up to 0.15). Hence, if we plot the energy E versus the chemical potential μ and zoom in to look only at a few adjacent data points, we may expect to see simply a straight line. If the points lie close enough, any deviation from linearity is less likely for physical reasons but rather due to different quality (i. e., proximity to the global minimum) of the approximation at the points. Hence, we can interpret slight deviations towards higher [lower] energy as an indication that the point is a worse [better] approximant than its neighbours. As the slope varies too little to clearly see these differences it is helpful to take the second numerical derivative to enhance the differences. Following the sketch in Fig. 8.12, a simple heuristic emerges: A pronounced peak in the second derivative means that the corresponding point is a better approximant than its neighbours. Hence, use its parameter vector as initial values to redo the minimisation at the neighbouring Hamiltonian parameters. If one of the two neighbours has a much lower second derivative than the other, redo only this one. Conversely, a point with a pronounced dip in the second derivatives should be re-done, starting with the parameter vector of one of its neighbours, normally the one with the higher second derivative.

Close to a phase transition, the procedure may get stuck because the step from one neighbour to the next changes the state too much. In this case, one should insert a new data points between the point that failed to get better and the neighbouring point used for the initial value.

8.6.4 Outlook to other minimization techniques

The literature on unconstrained nonlinear minimization is vast, and finding a good global minimization scheme requires a lot of trial and error. Apart from the two-phase heuristics described above, we have also tried genuine global minimization techniques, namely simulated annealing [KGV83] and differential evolution [SP97]. Both are genuinely global in the sense that they do not employ a local search stage. However, they thus cannot take advantage of the possibility to calculate the gradient. Hence, it is not surprising that simulated annealing converged much too slowly to be of use. (Simulated annealing is used in many different fields with much success but usually for functions with a convoluted potential surface but only few variables. We have several hundreds or even thousands of variables.) Differential evolution is a genetic algorithm and shows the —on first sight surprising— feature of converging to the mean field solution. (This seems explicable from the fact that crossing two genotypes in different basins has to end up in a "compromise", which is mean field.)

One further possibility might be basin hopping, which is a family of techniques (reviewed in [WS99]) that combine simulated annealing with a local search phase in order to overcome the problem states in the previous paragraph. These ideas are quite recent and research is still ongoing. So far, however, it seems that the basin hopping requires to perform very many local searches which hence have to converge fast. This is unfortunately not so in our case. It seems conceivable that variants can be developed that only use rough and hence fast local searches, and this might be a way to proceed with our method.

Another ansatz is using a clustering stage in the global phase of a two-phase method [RT87]. This allows to make multi-start much more efficient but has two difficult requirements: (i) One needs to factor any degeneracies in the minima out of the parameter space. We have not yet studied whether this is possible. (ii) The number of local minima must be small enough that one has a decent chance to encounter all of them during the local searches. Unfortunately, especially the calculations for Fig. 8.11 have brought us to the observation that the number of minima seems to grow very fast with the system size.

A further technique that we have tried is imaginary time evolution, which works as follows. Given an initial state $|\Psi(\mathbf{x}_0)\rangle$ chosen at random, we can find an estimate $|\Psi(\mathbf{x}_{i+1})\rangle \approx \mathrm{nrm}\, e^{-\Delta t H}\, |\Psi(\mathbf{x}_{i+1})\rangle$ for the discretized evolution of the state under the system Hamiltonian in the imaginary time direction. As for most initial states $|\Psi_0\rangle$, $\Psi_{\mathrm{G}} = \lim_{t\to\infty} \mathrm{nrm}\, e^{-tH}$ is the ground state, this iterative evolution should converge to a good approximation of the ground state. We have decomposed $e^{-\Delta t H}$ into a product of bond terms $e^{-\Delta t H_{ab}}$ using standard Trotter decomposition and then tried to find the $\Delta\mathbf{x} \in \mathbb{R}^K$ that maximizes the overlap

$$\left| \frac{\left\langle \Psi(\mathbf{x}+\Delta\mathbf{x}) \,\middle|\, e^{-\Delta t H_{ab}} \,\middle|\, \Psi(\mathbf{x}) \right\rangle}{\sqrt{\langle \Psi(\mathbf{x}+\Delta\mathbf{x}) \,|\, \Psi(\mathbf{x}+\Delta\mathbf{x}) \rangle \, \langle \Psi(\mathbf{x}) \,|\, e^{-2\Delta t H_{ab}} \,|\, \Psi(\mathbf{x}) \rangle}} \right| .$$

Unfortunately, the maximization failed to give good result even for arbitrarily small time steps Δt and we thus abandoned this approach.

We should also mention that for the some of the results of or previous paper [APD$^+$06], we (actually, M. Plenio, who programmed this part) have used a Rayleigh minimization technique: One restricts the energy function $E(\mathbf{x})$ in the sense that one keeps all but a few parameters fixed. For certain such subsets of only a few parameters, namely for the set of parameters corresponding to a single local unitary or to the phases and deformations for one pair of qubits, one can write the restricted energy function as quotient of two quadratic forms. This is also known as a generalized Rayleigh quotient and the global minimum can be find via a generalized eigenvalue problem. Such a "global minimum" typically is, however, not even a local minimum of the full energy function. The reason that we got good result for the Ising model in [APD$^+$06] seems now, in retrospect, have been due to the extraordinarily benign form of the corresponding energy landscape. Hence and because the scheme cannot easily generalized to spins higher than 1/2, we did not pursue this any further.

8.7 Conclusion and outlook

To conclude, we have presented a class of variational states that holds promise to approximate the ground states of spin systems and bosonic systems. The advantageous properties of this class is that it includes states with an arbitrarily high entanglement and the possibility to adapt to arbitrary geometries and number of spatial dimensions. We have shown how to calculate expectation values of observables for these states and demonstrated the approximation of the ground state for two model systems, namely the XY spin-1/2 model and the Bose-Hubbard model, in one and two dimensions. Furthermore, we have explained heuristics suitable to drive the minimization.

The method works for small systems and maps out the rough structure of phase diagrams. (The system sizes, though small, were sufficient to see the phase boundaries even though phase boundaries are defined, strictly speaking, only in the thermodynamical limit.) We can calculate observables for states in systems of considerable size but have problems in approximating the ground state in larger system to precision sufficient to see actual differences between different system sizes and hence to do finite-size scaling studies.

It seems likely that this is not because there were no states in our variational class which were close enough to approximate such ground states well. Rather, we simply cannot find them because our minimization gets trapped in local minima. Can the avoid this? This is the crucial question for the future development of the scheme, and at this moment, we may only offer some thoughts on that: It seems unlikely that the choice of another generic global minimization algorithm is able to steer around these local minima better that those algorithms that we have tried. For further progress, it seems hence desirable to have a better understanding of the shape and structure of the manifold $\Psi(\mathbb{R}^K) \subset \mathcal{H}$, i. e., our variational set of states as described by the mapping from the parameter space. Is, for example, this manifold "folded" more and curved stronger than the equi-energy surfaces of typical system Hamiltonians? This might explain, why there are so many local minima — and getting a better grasp on topology and metric of the mapping Ψ and its image could be most helpful in finding a better way to steer towards good minima.

Acknowledgements

We are indebted to G. Pupillo for providing the data from exact diagonalisation used in Fig. 8.9, and we would like to thank G. Pupillo, H.-P. Büchler and J. Eisert for helpful explanations on the Bose-Hubbard model. We thank M. B. Plenio and V. Verstraete for discussions. The numerical calculations presented in this work have been carried out using the compute cluster of the University of Innsbruck's Konsortium Hochleistungsrechnen. This work was supported by the Austrian Science Foundation (FWF) and the European Union (Projects OLAQUI and SCALA).

8.A Appendix A: Notes on the implementation

8.A.1 Avoiding overflows

A certain detail is worth mentioning as it may cause some difficulty in the implementation: As the product (8.15) contains $\mathcal{O}(N)$ terms, its values grows exponential with the system size N. Even for factors which are quite close to 1, the value will leave the range of floating-point arithmetics (on most computers, ca. $10^{-308} \ldots 10^{308}$) for even moderate values of N. To avoid this, one has to compute the product by summing up the *logarithms* of the matrix elements of $\rho^{jk}_{ab,c}$, then subtracting a constant from this sum, and then exponentiating the result component-wise. The subtraction of the constant does not change the final result, as it formally cancels against the final normalisation to unit trace. The exact value of the constant is hence irrelevant, but it has to be chosen large enough to avoid a floating-point overflow during exponentiation, but not so large that the elements of all the matrices ρ^{jk}_{ab} vanish due to floating-point underflows. (That some elements of some of the matrices ρ^{jk}_{ab} suffer an underflow is, however, unavoidable, but harmless, as their contribution to the result is evidently insignificant.) Especially for large systems, the constant has to be readjusted during the minimization.

8.A.2 Choice of programming languages

We have written two implementations of our algorithm. The first one, called "ewgs" is specialised for spin-1/2. It was used for the results on the XY model (Sec. 8.5.1), and also for the results presented in [APD$^+$06].[10] The other, more recent program is called "hwgs" and may be used for spins of any size n. "ewgs" is mainly written in C++, only the outer drivers are written in Python. Python [R$^+$] is a very modern, quite powerful scripting language, that features high-performance just-in-time compilation, an exceptionally comprehensive low- and high-level library, an open-source license and excellent inter-platform portability. The development of a numerics library for Python has reached maturity quite recently with the release of NumPy [Num, Oli06]. Due to the higher level of the language, development is much faster in Python than in C++. This makes it advisable to do most of the coding in Python and only write the "hot spots", i. e., the proverbial 10% of the code in which the processor spends 90% of the time, in an optimizing compiled language such as C++. This approach, though it may sound unusual to a traditionally oriented computer physicist, has been used in several places with much success (see e. g. the advocacy in [BCG05]), and from our experiences, we clearly recommend its use. Hence, for our second implementation, "hwgs", we followed this paradigm consequently and wrote only a small part in C++. This part was bound to the main Python code using SWIG [B$^+$]. For the local minimizer we used in both implementations the L-BFGS-B Fortran code [ZBLN97], linked to Python with the help of the tool f2py [Pet].

8.A.3 Performance

The performance of the "hwgs" implementation can be seen in Fig. 8.13. The blue curves shows the time required to calculate energy and full gradient

---

[10] For completeness, we should point out one difference between the description in this article and the implementation: In "ewgs", the unitaries are not parametrised using the Cayley transform, but rather as linear combination of the identity and the Pauli matrices: $U = \frac{u_0 \mathbb{1} + u_1 \sigma_x + u_2 \sigma_y + u_3 \sigma_z}{u_0^2 + u_1^2 + u_2^2 + u_3^2}$.

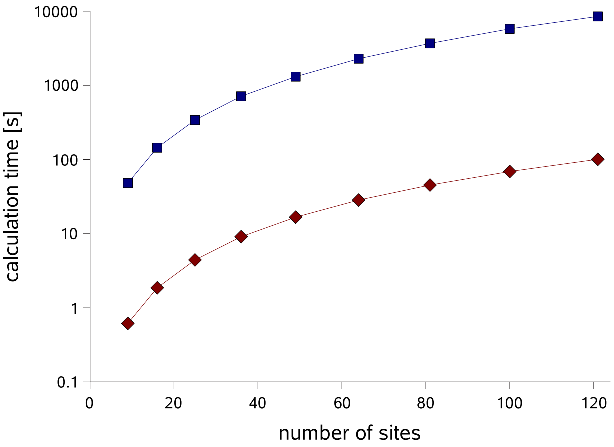


Figure 8.13: Performance of our "hwgs" implementation: For a Bose-Hubbard system on a square lattice of varying size, the calculation time for a single reduced density matrix of a pair of sites (red diamonds) and for the full gradient of the energy (derivatives w. r. t. *all* parameters) (blue squares) is shown. These calculations have been done for states with $n = 5$ levels per site and no superpositions ($m = 1$). The program was run on AMD Opteron machines clocked at 2.2 GHz.

for one parameter vector at various system sizes. In order to see the time required to find a good approximant, this has to be multiplied with the number of function evaluations needed by the minimiser.

Usually, one wants to find approximants for several different values of the Hamiltonian parameters. Then, one can save much time by running these minimisation in parallel if one has access to a computer cluster.

8.A.4 Availability

We would welcome to see our code been used in further projects. Hence, researchers who are interested in applying our code in their own projects are encouraged to contact the authors.

8.A.5 Density plots

The plotting technique used to obtain Figs. 8.4 and 8.7 merits a brief explanation. For these plots, we calculated the plotted quantity at different value pairs for the quantities at the x and y axes. In order to work out interesting feature, we did not evaluate at a fixed grid but rather started with some loosely spaced points to get an overview and then added more and more points at regions with interesting features. This allowed us to "explore" the parameter plane. However, it leaves us with a list of data points at irregular positions, which makes the usual 3D mesh plots unsuitable (as a mesh plot requires data from a regular grid). This is why we visualize the data instead with density plots, using colour to indicate z height. To obtain the colours we interpolated between the data point, and for this, we experimented with two interpolation algorithms, namely Akima's spline method [Aki96] and the Sibson's natural neighbours method [Sib81]. As the former has problems with strongly varying curvature (and this is the case here: the data varies more strongly near the phase transition than in the interiors of the phases) we used Sibson's method and produced Figs. 8.4 and 8.7 with the help of the NATGRID implementation [Cla04] of Sibson's algorithm.

8.B Appendix B: Calculating the gradient of the energy with respect to the parameter vector

For use in the gradient-based minimisation we need a fast way to obtain the gradient of the energy function $E(\mathbf{x})$. For the following, we assume that the Hamiltonian can be written in bond form,

$$H = \sum_{(a,b)\in\mathcal{B}} H_{ab}^{(ab)}.$$

As before, $\mathcal{B}$ is the set of bonds, i. e. of pairs of interacting spins. In many cases H_{ab} is the same for all bonds ab, but having an inhomogeneous Hamiltonian is no complication.

As the energy function is given by $E(\mathbf{x}) = \sum_{(a,b)\in\mathcal{B}} \operatorname{tr} H_{ab}\rho_{(a,b)}$, its gradient consists of a sum of derivatives of the reduced density matrices

$$\frac{\partial E(\mathbf{x})}{\partial x_l} = \sum_{(a,b)\in\mathcal{B}} \operatorname{tr} H_{ab} \frac{\partial \rho_{ab}}{\partial x_l}.$$

We shall now derive formulae for the components of the gradient, i. e., for the derivatives w. r. t. the different kinds of parameters.

8.B.1 Derivatives w. r. t. the parameters for the local unitaries

The derivative of a matrix exponential w. r. t. the components of the exponentiated matrix (or of linear combinations of these) is a very involved problem. Not only is the integral representation of this parametric derivative, though simple, in no way obvious, but also is the evaluation of this integral a very non-trivial matter. For a review of the history of this problem and current state of knowledge, consult Ref. [NH95].

For us, this is the main reason why we do not use the exponentiation of a Hermitean matrix for the parametrisation of the local unitaries, but rather the Cayley transform of it, for the latter involves only a matrix inverse, whose parametric derivative is expressed by a simple formula: For any invertible square matrix $A = A(t)$ that depends differentiably on a real parameter t, we have

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}t} = -A^{-1}\frac{\mathrm{d}A}{\mathrm{d}t}A^{-1} \tag{8.23}$$

(for a proof, see e. g. [PlM]).

We need n^2 real parameters to parametrise a Hermitean $n \times n$ matrix, which we arrange to form a $n \times n$ upper triangular matrix $\tilde{A}$ with real entries in the diagonal, complex entries in the upper triangle and zeroes in the lower triangle. $A = \tilde{A} + \tilde{A}^\dagger$ is now Hermitean and

$$U = \left(i\mathbb{1} + \tilde{A} + \tilde{A}^\dagger\right)\left(i\mathbb{1} - \tilde{A} - \tilde{A}^\dagger\right)^{-1} \tag{8.24}$$

is unitary. Using Eq. (8.23), we get

$$\begin{aligned}
\frac{\partial U}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}A_{kl}} &= \left\{\begin{matrix}1\\ i\end{matrix}\right\}(\mathbb{1}+U)\left(|k\rangle\langle l| \left\{\begin{matrix}+\\ -\end{matrix}\right\} |l\rangle\langle k|\right)(i\mathbb{1}-A)^{-1}, \qquad (8.25)\\
\frac{\partial U^\dagger}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}A_{kl}} &= -\left\{\begin{matrix}1\\ i\end{matrix}\right\}(i\mathbb{1}+A)^{-1}\left(|k\rangle\langle l| \left\{\begin{matrix}+\\ -\end{matrix}\right\} |l\rangle\langle k|\right)(\mathbb{1}+U^\dagger)\\
&= \left(\frac{\partial U}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}A_{kl}}\right)^\dagger.
\end{aligned}$$

We use this to calculate

$$\begin{aligned}
\frac{\partial E}{\partial A_{a,kl}} = \sum_{b:(b,a)\in\mathcal{B}} \operatorname{tr} H_{ba}\Bigg[&\left(U_b \otimes \frac{\partial U_a}{\partial A_{a,kl}}\right)\tilde{\tilde{\rho}}_{ba}\,(U_b\otimes U_a)^\dagger +\\
&+ (U_b\otimes U_a)\,\tilde{\tilde{\rho}}_{ba}\left(U_b\otimes\frac{\partial U_a}{\partial A_{a,kl}}\right)^\dagger\Bigg] +\\
+ \sum_{c:(a,c)\in\mathcal{B}} \operatorname{tr} H_{ac}\Bigg[&\left(\frac{\partial U_a}{\partial A_{a,kl}}\otimes U_c\right)\tilde{\tilde{\rho}}_{ac}\,(U_a\otimes U_c)^\dagger +\\
&+ (U_a\otimes U_c)\,\tilde{\tilde{\rho}}_{ac}\left(\frac{\partial U_a}{\partial A_{a,kl}}\otimes U_c\right)^\dagger\Bigg],
\end{aligned}$$

where $\tilde{\tilde{\rho}}_{bc}$ is the reduced density matrix without application of the local unitaries, i. e. $\tilde{\tilde{\rho}}_{bc} = (U_b\otimes U_c)^\dagger\rho_{bc}(U_b\otimes U_c)$.

8.B.2 Derivatives w. r. t. the deformation parameters

For the derivatives w. r. t. the parameters $\operatorname{Re} d^l_{c,s}$ and $\operatorname{Im} d^l_{c,s}$, we have to take care of the normalisation of ρ_{ab} as it depends on those parameters. We abbreviate the middle line of Eq. (8.12) with $\tilde{\rho}_{ab}$ and start with using Eq. (8.19) in order to see that

$$\tilde{\rho}_{ab} = \sum_{j,k=1}^{m} \alpha_j \alpha_k^* \check{D}_{ab}^{jk} \odot \rho_{ab}^{jk}$$

where (according to Eq. (8.20))

$$\check{D}_{ab}^{jk} := \sum_{\mathbf{r},\mathbf{r}' \in \mathbb{S}} d_{a,r_1}^{a} d_{a,r_2}^{b} d_{a,r_1'}^{a\ *} d_{a,r_2'}^{b\ *} \left|\mathbf{r}\right\rangle \left\langle \mathbf{r}'\right| .$$

We write the derivative as

$$\frac{\partial \rho_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{c,s}^{l}} = (U_a \otimes U_b) W_{\varphi_{ab}} \left(\frac{\partial}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{c,s}^{l}} \frac{\tilde{\rho}_{ab}}{\operatorname{tr} \tilde{\rho}_{ab}} \right) W_{-\varphi_{ab}} (U_a \otimes U_b)^{\dagger}, \tag{8.26}$$

where the term in parentheses becomes

$$\frac{\frac{\partial \tilde{\rho}_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{c,s}^{l}} \operatorname{tr} \tilde{\rho}_{ab} - \tilde{\rho}_{ab} \operatorname{tr} \frac{\partial \tilde{\rho}_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{c,s}^{l}}}{\left(\operatorname{tr} \tilde{\rho}_{ab}\right)^2} \tag{8.27}$$

In order to evaluate $\partial \tilde{\rho}_{ab} / \partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_c^l$, we distinguish three cases, namely (i) $c = a$, (ii) $c = b$, (iii) $c \notin \{a, b\}$.

Case (i): The only term in the middle line of Eq. (8.12) that depends on d_a^l is $\check{D}_{ab}^{jk}$ and this only for those terms in the sum, where $j = l$ or $k = l$. One finds

$$\frac{\partial \tilde{\rho}_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{as}^{l}} = \left\{\begin{matrix}1\\ i\end{matrix}\right\} \alpha_l \sum_{k=1}^{m} \alpha_k^* \sum_{r_1, r_1', r_2' \in \mathbb{S}} d_{br_1}^{l} d_{ar_1'}^{k\ *} d_{br_2'}^{k\ *} \left|r_1 s\right\rangle \left\langle r_1' r_2'\right| \odot \rho_{ab}^{lk} + \text{H.c.}$$

Case (ii): Analogous:

$$\frac{\partial \tilde{\rho}_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{bs}^{l}} = \left\{\begin{matrix}1\\ i\end{matrix}\right\} \alpha_l \sum_{k=1}^{m} \alpha_k^* \sum_{r_2, r_1', r_2' \in \mathbb{S}} d_{ar_2}^{l} d_{ar_1'}^{k\ *} d_{br_2'}^{k\ *} \left|s r_2\right\rangle \left\langle r_1' r_2'\right| \odot \rho_{ab}^{lk} + \text{H.c.},$$

Case (iii): For $c \notin \{a, b\}$, $\check{D}_{ab}^{jk}$ is independent of d_c^l, but ρ_{ab}^{jk} is now dependent. We get:

$$\begin{aligned} \frac{\partial \tilde{\rho}_{ab}}{\partial \left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\} d_{cs}^{l}} = {} & \left\{\begin{matrix}1\\ i\end{matrix}\right\} \alpha_l \sum_{k=1}^{m} \alpha_k^* \check{D}_{ab}^{lk} \odot \\ & \odot d_{cs}^{k*} \left(\exp i \left(\Phi_{ac}^{r_1 s} - \Phi_{ac}^{r_1' s} + \Phi_{bc}^{r_2 s} - \Phi_{bc}^{r_2' s} \right) \right)_{\mathbf{r},\mathbf{r}' \in \mathbb{S}^2} \odot \bigodot_{e \in V \setminus \{a,b,c\}} \rho_e^{lk} + \text{H.c.}, \end{aligned}$$

where ρ_e^{lk} is given by Eq. (8.18).

8.B.3 Derivatives w. r. t. the superposition coefficients

The derivatives w. r. t. $\operatorname{Re}\alpha_l$ and $\operatorname{Im}\alpha_l$ are found the same way as for the deformation, and one gets

$$\frac{\partial \operatorname{nrm}\tilde{\rho}_{ab}}{\partial \alpha_{ab}} = \frac{\frac{\partial\tilde{\rho}_{ab}}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}\alpha^l}\operatorname{tr}\tilde{\rho}_{ab} - \tilde{\rho}_{ab}\operatorname{tr}\frac{\partial\tilde{\rho}_{ab}}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}\alpha_l}}{\left(\operatorname{tr}\tilde{\rho}_{ab}\right)^2}$$

with

$$\frac{\partial\tilde{\rho}_{ab}}{\partial\left\{\begin{smallmatrix}\mathrm{Re}\\ \mathrm{Im}\end{smallmatrix}\right\}\alpha_l} = \left\{\begin{matrix}1\\ i\end{matrix}\right\}\sum_{k=1}^{m}\alpha_k^*\left|D^l_{(a,b)}\right\rangle\left\langle D^k_{(a,b)}\right|\odot\tilde{\rho}^{lk}_{ab} + \text{H.c.},$$

where $\left|D^l_{(a,b)}\right\rangle$ is defined in Eq. (8.20).

8.B.4 Derivatives w. r. t. the phases

We first introduce

$$\check{W}_\Phi = |\Phi\rangle\langle\Phi| \quad \text{with } |\Phi\rangle = \sum_{\mathbf{r}\in\mathrm{S}^2}|\mathbf{r}\rangle\, e^{i\Phi^{r_1r_2}},$$

so that we can write —due to Eq. (8.21)— Eq. (8.12) as

$$\rho_{ab} = (U_a\otimes U_b)\left(\check{W}_{\Phi_{ab}}\odot\tilde{\rho}_{ab}\right)(U_a\otimes U_b).$$

We have to take care that in case of a symmetrisation according to Sec. 8.3.4 a phase can occur more than once in the expression for ρ_{ab}, and hence, we make use of the phase index mapping $\nu(a,b)\in\{1,\dots,R\}$ (Sec. 8.3.4), that associates with every pair of spins a, b a phase matrix $\Phi_{\nu(a,b)}$. We write (with $\mathbf{r}\equiv(r_1r_2)$)

$$\frac{\partial\rho_{ab}}{\partial\Phi^{\mathbf{r}}_\nu} = \frac{1}{\operatorname{tr}\rho_{ab}}(U_a\otimes U_b)\left(\frac{\partial\check{W}_{\Phi_{\nu(a,b)}}}{\partial\Phi^{\mathbf{r}}_\nu}\odot\tilde{\rho}_{ab} + \check{W}_{\Phi_{\nu(a,b)}}\odot\frac{\tilde{\rho}_{ab}}{\partial\Phi^{\mathbf{r}}_\nu}\right)(U_a\otimes U_b)^\dagger$$

and proceed to discuss the two derivatives in this expression.

The first one is evidently non-zero only if $\nu=\nu(a,b)$ and then evaluates to

$$\frac{\partial\check{W}_{\Phi_{\nu(a,b)}}}{\partial\Phi^{\mathbf{r}}_{\nu(a,b)}} = i\check{W}_{\Phi_{\nu(a,b)}}\odot\left(\delta_{\{q_1,q_2\},\{r_1,r_2\}} - \delta_{\{q_1',q_2'\},\{r_1,r_2\}}\right)_{\mathbf{q},\mathbf{q}'\in\mathrm{S}^2}.$$

The set notation in the Kronecker deltas accounts for the fact that Φ is symmetrised, $\Phi^{r_1r_2}=\Phi^{r_2r_1}$, (cf. again Sec. 8.3.4) and hence, the order of the components of the vectors $\mathbf{r}$, $\mathbf{q}$, and $\mathbf{q}'$ must be disregarded.

For the second term, we pull the derivative inwards

$$\frac{\partial\tilde{\rho}_{ab}}{\partial\Phi^{\mathbf{r}}_\nu} = \sum_{j,k=1}^{m}\alpha_j\alpha_k^*\check{D}^{jk}_{ab}\odot\frac{\partial\rho^{jk}_{ab}}{\partial\Phi^{\mathbf{r}}_\nu},$$

then rewrite Eq. (8.15) as

$$\rho_{ab}^{jk} = \bigodot_{c \in V \setminus \{a,b\}} \sum_{s \in \mathbb{S}} d_{cs}^{j} d_{cs}^{k\,*} \left|\tilde{\Phi}_{abc}^{s}\right\rangle \left\langle\tilde{\Phi}_{abc}^{s}\right|$$

with

$$\left|\tilde{\Phi}_{abc}^{s}\right\rangle = \sum_{q_1,q_2 \in \mathbb{S}} |q_1 q_2\rangle \exp\left[i\left(\Phi_{\nu(a,c)}^{q_1 s} + \Phi_{\nu(b,c)}^{q_2 s}\right)\right]$$

and continue

$$\begin{aligned}\frac{\partial \rho_{ab}^{jk}}{\partial \Phi_{\nu}^{\mathbf{r}}} = \sum_{e \in V \setminus \{a,b\}} \left(\bigodot_{c \in V \setminus \{a,b,e\}} \sum_{s \in \mathbb{S}} d_{cs}^{j} d_{cs}^{k\,*} \left|\tilde{\Phi}_{abc}^{s}\right\rangle \left\langle\tilde{\Phi}_{abc}^{s}\right| \right) \odot \\ \odot \sum_{s \in \mathbb{S}} d_{es}^{j} d_{es}^{k\,*} \frac{\partial \left|\tilde{\Phi}_{abe}^{s}\right\rangle \left\langle\tilde{\Phi}_{abe}^{s}\right|}{\partial \Phi_{\nu}^{\mathbf{r}}}.\end{aligned}$$

The sum over e formally runs over $N-2$ terms. Most of these vanish, however, namely all those for which neither $\nu = \nu(a,e)$ nor $\nu = \nu(b,e)$. For translation-invariant phases, the number of remaining terms is of the order of the coordination number of the lattice.

The derivative in the last line of the previous equation evaluates to

$$\begin{aligned}\frac{\partial \left|\tilde{\Phi}_{abe}^{s}\right\rangle \left\langle\tilde{\Phi}_{abe}^{s}\right|}{\partial \Phi_{\nu}^{r_1 r_2}} = i \left|\tilde{\Phi}_{abe}^{s}\right\rangle \left\langle\tilde{\Phi}_{abe}^{s}\right| \odot \\ \odot \Big(\delta_{\nu,\nu(a,e)} \left[\delta_{\{r_1,r_2\},\{q_1,s\}} - \delta_{\{r_1,r_2\},\{q_1',s\}}\right] + \\ + \delta_{\nu,\nu(b,e)} \left[\delta_{\{r_1,r_2\},\{q_2,s\}} - \delta_{\{r_1,r_2\},\{q_2',s\}}\right]\Big)_{\mathbf{q},\mathbf{q}' \in \mathbb{S}^2}.\end{aligned}$$

Chapter 9

Tensor networks

This chapter is concerned with a class of states known as tensor network states and a sub-class thereof, the tensor tree states. These states are a generalization of the matrix-product states discussed in Sec. 6.4, and we hence continue in the context laid out in Sec. 6.4.4.

We shall start with a precise definition of tensor network states and then concentrate on the case where these networks are trees. In Ref. [MRS02] (see also [Rod02]), a scheme called "DMRG for Trees" is used to study excitons on dendrimers.[1] While their ansatz required that the interaction pattern of the studied system has the form of a tree, the idea can be generalized to arbitrary systems, as shown in the "tensor tree network" ansatz of Shi et al. [SDV06]. As we shall see, the advantage of tree states is that their expectation value for any product observable can be calculated exactly by means of a "contraction", in contrast to states with an underlying mesh structure such as the projected entangled pairs states discussed in Sec. 6.4.4. Markov and Shi [MS07] have studied when a tensor network can be contracted exactly and have given a criterion on how far one may stray from the tree shape without losing this ability, but we shall not pursue this point.

The work presented in this section is (yet unpublished) work done in collaboration with Robert Hübener, Caroline Kruszynska, Lorenz Hartmann, Wolfgang Dür, and Hans J. Briegel. The presentation in this chapter focuses on my parts in this work and does not go into much detail of certain aspects investigated by my colleagues.

9.1 Definition

Let us first define the notion of tensor networks in a very general form: A *tensor network* is a connected (mathematical) graph such that with each edge $\{a, b\}$ is associated a positive integer $\chi^{[a,b]}$, which we call the *edge range*, and with each vertex a of degree $\eta > 1$ a complex-valued tensor $T^{[a]}$ of rank η

---

[1] For a review on dendrimers, see [Ino00]

is associated, where the ranges of the η indices of $T^{[a]}$ are given by the edge ranges. This means that for a vertex a, which is connected to the vertices $b_1, b_2, \ldots, b_\eta$, the ranges of the indices of $T^{[a]}$ are

$$0, \ldots, \chi(a, b_1) - 1, \quad 0, \ldots, \chi(a, b_2) - 1, \quad \ldots, \quad 0, \ldots, \chi(a, b_\eta) - 1$$

and $T^{[a]}$ has $\prod_{i=1}^{\eta} \chi(a, b_i)$ complex entries. Only the vertices with degree $\eta > 1$ have associated tensors and these are called the *tensor vertices*, while we call the vertices with degree 1 the *leaf vertices*. An edge between two tensor vertices shall be called an *inner edge*, one leading to a leaf vertex a *leaf edge*. Note that an inner edge connects two tensors, and each of them has one index corresponding to this edge. The ranges of these two indices are the same (namely the edge range), and hence, we may contract along such an edge.

A tensor network describes a quantum state by means of the following convention: With each leaf vertex a we associate a d-level quantum system, where d is the range of the edge incident onto a. (Remember that leaf vertices do not carry tensors. Hence, the edge range of the edge incident on the leaf vertex may be chosen such that it matches the dimensionality of the system we wish to describe, and the tensor at the other end of the edge then needs to have its corresponding index to have the same range.) Thus, if we have a tensor network with M tensor vertices with associated tensors $T^{[1]}, \ldots, T^{[M]}$ and N leaf vertices $a_1, \ldots, a_N$ with corresponding edge ranges $d_1, \ldots, d_N$, we describe a quantum state living in the Hilbert space

$$\mathcal{H} = \bigotimes_{i=1}^{N} \mathbb{C}^{d_i}.$$

This state is now formally written as

$$|\mathrm{TN}\rangle = \sum_{s_1=0}^{d_1-1} \cdots \sum_{s_N=0}^{d_N-1} \mathcal{F}\left(T^{[1]}, \ldots, T^{[M]}; s_1, \ldots, s_N\right) |s_1, \ldots, s_N\rangle . \tag{9.1}$$

The operator $\mathcal{F}$ carries out a contraction as follows: Each leaf vertex a_i is connected by a leaf edge to one tensor vertex $\tilde{b}$ with tensor $T^{[\tilde{b}]}$, and this tensor has an index corresponding to the leaf edge. All such indices are fixed to the values s_i. The remaining indices of all the tensors correspond to inner edges, over which a contraction is carried out. Carrying out these contractions will result in a single scalar, namely the coefficient $\langle s_1, \ldots, s_N \,|\, \mathrm{TN}\rangle$ of the computational basis expansion of $|\mathrm{TN}\rangle$.

An example should make this clearer: Consider the tensor network in Fig. 9.1. It corresponds to the state

$$|\psi\rangle = \sum_{s_1=0}^{d_1-1} \cdots \sum_{s_5=0}^{d_5-1} |s_1 s_2 s_3 s_4 s_5\rangle \sum_{\alpha^{[1,2]}=0}^{\chi^{[1,2]}} \sum_{\alpha^{[2,3]}=0}^{\chi^{[2,3]}} T^{[1]}_{s_1 s_2 \alpha^{[1,2]}} T^{[2]}_{s_3 \alpha^{[1,2]} \alpha^{[2,3]}} T^{[3]}_{s_4 s_5 \alpha^{[2,3]}}.$$

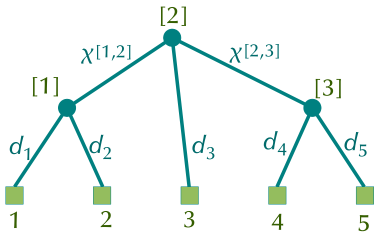


Figure 9.1: An example for a tensor network. Here, the graph is a tree. Squares are leaf vertices and circles represent tensor vertices, both labelled with their respective vertex indices. For tensor vertices, we put square brackets around the index. The edges are labelled with their edge ranges, where we use the convention to write χ for inner edges and d for leaf edges.

9.2 Tensor trees

In this section we explore how tensor *trees* can be treated numerically, as this is the ansatz that is currently being researched in our group.

The graph of Fig. 9.1 is a tree, i e., a connected graph without loops. This special structure allows us to calculate for the associated state $|\psi\rangle$ exactly and efficiently the expectation value of any observable of the product form

$$O = \bigotimes_{i=1}^{N} O_i. \tag{9.2}$$

Consequently, the same is true for sums of such product observables. Staying with the example of Fig. 9.1 we write (using now α for $\alpha^{[1,2]}$ and β for $\alpha^{[2,3]}$ to make the equation more readable)

$$\begin{aligned}\langle \psi \,|\, O \,|\, \psi \rangle = \sum_{s_1', s_1 = 0}^{d_1 - 1} \cdots \sum_{s_5', s_5 = 0}^{d_5 - 1} \left(\prod_{i=1}^{5} \langle s_i' \,|\, O_i \,|\, s_i \rangle \right) \times \\ \times \sum_{\alpha', \alpha = 0}^{\chi^{[1,2]}} \sum_{\beta', \beta = 0}^{\chi^{[2,3]}} T^{[1]*}_{s_1' s_2' \alpha'} T^{[1]}_{s_1 s_2 \alpha} T^{[2]*}_{s_3' \alpha' \beta'} T^{[2]}_{s_3 \alpha \beta} T^{[3]*}_{s_4' s_5' \beta'} T^{[3]}_{s_4 s_5 \beta}\end{aligned} \tag{9.3}$$

At first glance, this expression is a sum over exponentially many terms. However, we can rearrange terms by splitting the tree into subtrees. For example, if we split along the inner edge [2,3], the sums over β' and β run over products of two factors, each of which corresponds to one subtree and can be calculated independently of the other:

$$\langle \psi \,|\, O \,|\, \psi \rangle = \sum_{\beta', \beta = 0}^{\chi^{[2,3]} - 1} B^{[2]}(\beta', \beta) B^{[3]}(\beta', \beta).$$

Here, the factor corresponding to the subtree below tensor vertex [3] is

$$B^{[3]}(\beta', \beta) = \sum_{s_4', s_4=0}^{d_4-1} \sum_{s_5', s_5=0}^{d_5-1} T^{[3]*}_{s_4' s_5' \beta'} T^{[3]}_{s_4 s_5 \beta} \left\langle s_4' \,\middle|\, O_4 \,\middle|\, s_4 \right\rangle \left\langle s_5' \,\middle|\, O_5 \,\middle|\, s_5 \right\rangle$$

and the factor for the subtree below tensor vertex [2] is again divided up into factors corresponding to the subtree below tensor vertex [1] and to spin 3

$$B^{[2]}(\beta', \beta) = \sum_{\alpha', \alpha=0}^{\chi^{[1,2]}-1} \sum_{s_3', s_3=0}^{d_3-1} T^{[2]*}_{s_3', \alpha', \beta'} T^{[2]}_{s_3, \alpha, \beta} B^{[1]}(\alpha', \alpha) \left\langle s_3' \,\middle|\, O_3 \,\middle|\, s_3 \right\rangle,$$

and the last subtree is evaluated as

$$B^{[1]}(\alpha', \alpha) = \sum_{s_1', s_1=0}^{d_1-1} \sum_{s_2', s_2=0}^{d_2-1} T^{[1]*}_{s_1' s_2' \alpha'} T^{[1]}_{s_1 s_2 \alpha} \left\langle s_1' \,\middle|\, O_1 \,\middle|\, s_1 \right\rangle \left\langle s_2' \,\middle|\, O_2 \,\middle|\, s_2 \right\rangle.$$

In practice, we have to work from bottom to top: We first compute $B^{[1]}$, from this $B^{[2]}$ and combine the result with $B^{[3]}$ to get $\langle \psi \,|\, O \,|\, \psi \rangle$.

Let us define a *tensor tree state* to be any state associated with a root-less binary tree graph. By a root-less binary tree we mean a loop-less graph (such as the one in Fig. 9.1) with all vertices being either of degree 3 (tensor vertices) or 1 (leaf vertices).[2] Then, there are as many B terms as there are tensors, and in order to calculate one of them, we have to carry out a quadruple sum of $O(\chi^4)$ terms, where χ is the maximum of all the χ_i and d_i. Since a root-less binary tree with N leaves has $N-2$ tensor vertices, the time to calculate an expectation value of a product observable for a tensor tree state scales as $O(N\chi^4)$. These facts have been pointed out first in Ref. [MS07] and more explicitly (though using another notation) in Ref. [SDV06].

To implement this contraction numerically in an elegant fashion, one should use recursion:[3] First, we choose a *root edge*, i. e. an edge through which to make the first split. In the example above, we had chosen [2,3] as root edge. Then, we call the function calculate_observable, which is given in pseudo-code in Table 9.1. This function then calls calculate_vertex_value (Table 9.2), which takes three arguments, a vertex index b and two edge

---

[2] In graph theory, a graph is called a *tree* if it is connected and has no loops. It is called a binary tree if each vertex has either degree 3 or degree 1 (the latter are called *leaves*) except for one vertex that has degree 2 and is called the *root*. By convention, "trees grow downwards", i. e., they are always drawn with the root edge on top and the leaves at the bottom. Hence, a tree in graph theory does not have a height as natural trees do, but a *depth*, which is the maximum distance that a leaf has from the root. In our case, it turns out to simplify the formalism if one does not have a root vertex, and hence, our trees are "root-less". An alternative term to denote this kind of tree is Cayley tree of degree 3. (A Cayley tree of degree d is a tree that only contains vertices of degree d or degree 1.)

[3] A reader unfamiliar with the standard algorithms and techniques to treat trees in computer programming might find the relevant chapters of the classic textbook [Sed88] helpful.

```
function calculate_observable (root_edge $(b_l, b_r)$)
    $v \leftarrow 0$
    for $i'$ from 0 to $\chi(b_l, b_r)$
        for $i$ from 0 to $\chi(b_l, b_r)$
            $v \leftarrow v +$ calculate_vertex_value $(b_l, i', i) \times$
                $\times$ calculate_vertex_value $(b_r, i', i)$
        end for
    end for
    return $v$
end function
```

Table 9.1: The function calculate_observable in pseudo-code notation.

indices i', i. Its purpose is to calculate, when called for a tensor vertex b, the value $B^{[b]}(i', i)$ (by means of recursively calling itself), and when called for a leaf vertex b, the value $\langle i' \,|\, O_b \,|\, i \rangle$.

9.3 Other variants of tensor networks

Since Östlund and Rommer's realization of the connection between DMRG and matrix product states (see Sec. 6.4.2), a host of variations on this theme have been studied, as we have already seen in the overview given in Sec. 6.4.4. All these state classes can be conveniently described in a unified fashion in the language of tensor networks. We shall see that they fit into the general form of the definition given in Sec. 9.1. However, not for all possible tensor networks, observables can be calculated efficiently; we require a tree-structure for the scheme described in Sec. 9.2 to work, and a deviation from this tree structure is only possible up to a certain extent [MS07], unless one is willing to use approximative techniques.

9.3.1 Matrix product states

It is easy to see that matrix product states fit into the definition of a tensor network state: While the quantity $A^{[a]s_a}$ associated with a site a in Eqs. (6.1, 6.2) is usually seen as a tuple of d matrices, each of size $\chi \times \chi$, one can, of course, as well consider $A^{[a]}$ as one tensor of shape $\chi \times \chi \times d$. (Note that we now use χ to denote the quantity called D before.) Then, the matrix multiplication is exactly the same operation as the contraction $\mathcal{F}$ associated with the graphs depicted in Fig. 9.2.

In a matrix product state with periodic boundary conditions, all tensors have rank 3, and consequently, all tensor vertices in Fig. 9.2b have degree 3. In matrix product state with open boundary condition, however, the tensors

```
function calculate_vertex_value (vertex $b$, index $i'$, index $i$):
  if $b$ is a leaf vertex then
      return $\langle i' \,|\, O_b \,|\, i \rangle$
  else (i. e., if $b$ is a tensor vertex)
      if, during the calculation for the current observable, this function
              has already been called with the same arguments as now
      then
          return the values calculated last time by looking
              them up in the cache
      else
          Denote the 3 vertices adjacent to $b$ as follows:
              The vertex closest to the root edge is $b_u$, the other two
              vertices are (arbitrarily but consistently) $b_l$ and $b_r$.
          In the following, read the tensor elements $T^{[b]}_{pqr}$ such that the
              first index ($p$) corresponds to the edge to $b_u$, the second one
              ($q$) to the edge to $b_l$ and the third one ($r$) to the edge to $b_r$.
          $v \leftarrow 0$
          for $j'$ from 0 to $\chi(b, b_l)$
              for $j$ from 0 to $\chi(b, b_l)$
                  for $k'$ from 0 to $\chi(b, b_r)$
                      for $k$ from 0 to $\chi(b, b_r)$
                          $v \leftarrow v + T^{[b]*}_{i'j'k'} T^{[b]}_{ijk} \times$
                                $\times$ calculate_vertex_value $(b_l, j', j)$ $\times$
                                $\times$ calculate_vertex_value $(b_r, k', k)$
                      end for
                  end for
              end for
          end for
          Store $v$ in the cache as result for the arguments $(b, i', i)$.
          return $v$
      end if
  end if
end function
```

Table 9.2: The function calculate_vertex_value in pseudo-code notation.

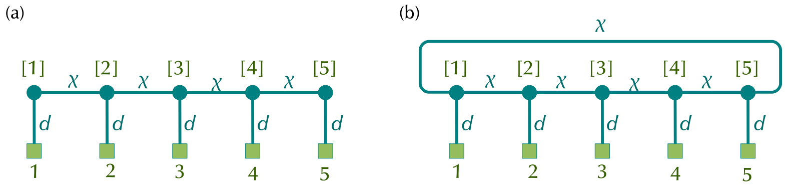


Figure 9.2: Tensor networks associated with a 5-spin matrix product state (spin dimension d, matrix dimension $\chi \times \chi$) with (a) open and (b) periodic boundary conditions. As in Fig. 9.1 before, squares indicate spin vertices (indexed with plain figures), and circles tensor vertices (indexed with figures in square brackets). Edges are labelled with their edge range.

at the end have only rank 2 and shape $\chi \times d$. In the original formulation of Eq. (6.1), this has been reflected by writing the quantities controlling the end spins as d-tuples of χ-dimensional vectors.

We see immediately that the open-boundary matrix product state is a tensor tree state, quite similar to the tree in Fig. 9.1. The difference is that the lowest tensors in Fig. 9.1 each have two leaf edges while in Fig. 9.2b, they have one leaf edge.

Note that we can efficiently evaluate an observable for a periodic-boundary matrix product state (using Eq. (6.3)) although the associated graph is not a tree, and the presence of a loop makes a contraction with the algorithm of Sec. 9.2 impossible. This is because a contraction is still possible even in the presence of loops in a limited sense. This is explored in detail in [MS07].

9.3.2 Projected entangled-pair states

In [VC04a] Verstraete and Cirac have proposed a way to generalize matrix product state to the 2D case inspired by the idea of valence-bond states (whose possible uses for quantum information they had studied before [VC04b]). Fig. 9.3 shows such a so-called projected-entangled pair state (PEPS) on a periodic 2D square lattice. Each tensor (depicted in the figure, as before, by a round bullet) now has five indices: One index specifies the spin level (represented in the figure by the edge to the square leaf) and hence runs from 0 to $d-1$, the other four indices connect to the four neighboring tensors and run from 0 to $\chi - 1$, where χ is, as before, the edge range (denoted D in Verstraete and Cirac's papers). As in the matrix product states, each tensor controls precisely one spin to which it is directly connected. Note that this is not the case for the tensor tree states. The one-to-one correspondence of tensors and spins allows for an alternative understanding of the constructing

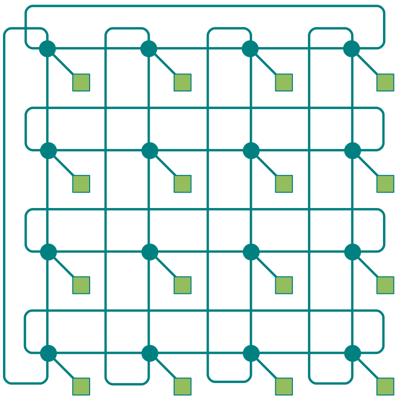
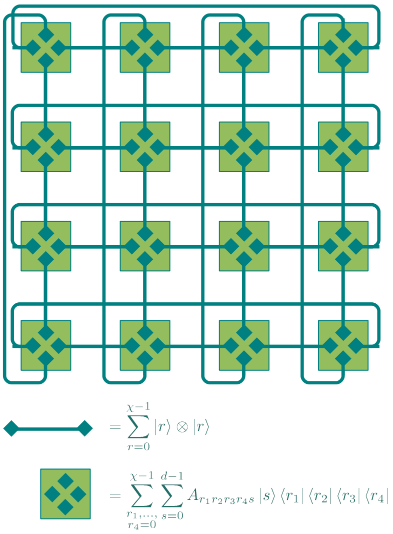


Figure 9.3: Tensor networks associated with a PEPS on a 4 × 4 lattice with periodic boundary conditions.



$= \sum_{r=0}^{\chi-1} |r\rangle \otimes |r\rangle$
$= \sum_{r_1,\ldots,r_4=0}^{\chi-1} \sum_{s=0}^{d-1} A_{r_1 r_2 r_3 r_4 s} |s\rangle \langle r_1| \langle r_2| \langle r_3| \langle r_4|$

Figure 9.4: Construction of the same state as in Fig. 9.3 in the valence bond picture.

of the states, namely in the picture of valence-bond states:[4] Each bond of the square lattice is modelled by a pair of maximally entangled states of Schmidt rank χ (see Fig. 9.4). At each site, four such auxiliary χ-level spins can now be found, each being one end of the pairs forming the four incident edges. The tensor $A^{[a]}$ associated with a site a is seen as a projector

$$\sum_{s=0}^{d-1} \sum_{\substack{r_1,r_2,\\ r_3,r_4=0}}^{\chi} A^{[a]}_{sr_1r_2r_3r_4} \left|s\right\rangle \left\langle r_1r_2r_3r_4\right|$$

that maps down from the four-spin auxiliary space $(\mathbb{C}^\chi)^{\otimes 4}$ to the space $\mathbb{C}^d$ of the physical spin. (In the case of open boundary conditions, the tensors at the lattice edges have one index less, and at the corners, two indices less.) The value of the construction is that it immediately gives an upper bound on the block-wise entanglement of a PEPS with respect to any bipartition. As a maximally entangled state of two χ-level systems has an entropy of entanglement of $\log_2 \chi$, each bond that crosses the boundary area of the bipartition can contribute this amount to the block-wise entanglement. Hence, the entanglement with respect to any bipartition of the lattice that a PEPS can have is bounded from above by $\log_2 \chi$ times the number of bonds that cross the boundary area, and this bound can be saturated. As we shall see in the next section (Sec. 9.4), this property is crucial for making the PEPS a class of states suitable for 2D and 3D systems.

This advantage is bought by a considerably more complicated procedure to evaluate observables. Because of the many loops inherent in the mesh underlying a PEPS any straight-forward contraction in the sense of the procedure of Sec. 9.2 (or the more general procedure of Ref. [MS07]) is not possible. A simple way to contract the network is as follows: If one contracts an edge, two tensors, which both have further indices, will be joined to one tensor, which now may have up to 10 indices.[5] The tensor network now contains two triangles, and contracting an edge of such a triangle cause the other two edges to come to lie on top of each other, i.e., the two edges of rank χ become one edge of rank χ^2. Then, one truncates the two tensors joined by this edge by means of a singular value decomposition in order to reduce the

---

[4] This class of states were generalized from the expression used to describe the so-called valence-bond solid phase of certain antiferromagnetic spin systems [AKLT87]. They were used as a conceptual aid to understand the entanglement properties of matrix product states in [VPC04], and are also handy as an alternative construction principle for graph states, cf. Sec. 5.2.1.

[5] When calculating the expectation value $\langle\psi|O|\psi\rangle$ of an observable O, each tensor appears twice, once in the ket $|\psi\rangle$ and once as its complex conjugate in the bra $\langle\psi|$. When carrying out the double sum associated with the bra and ket states of the spin connected to the tensor, the edge range of the inner edges rises from χ to χ^2. Alternatively, if one starts by contracting an inner edge between two tensors first, one is left with a tensor with six inner edges (each of the original tensors has three further neighbors) and four leaf edges (the spins of both tensors, each as bra and as ket spin). This results in a tensor with $\chi^6 d^4$ elements.

edge dimension to χ. In this way, one contracts the whole network, always using singular value decompositions to truncate any edge which has been "blown up". However, as numerical experiments done by V. Murg and I. Cirac [pers. comm.] have shown, this procedure is not only computationally rather expensive but also and especially fails to reach satisfactory precision. The approximation error due to the repeated local truncations adds up to fast.

Hence, they have devised a more sophisticated scheme [VC04a, MVC07], which, however, works well only for open boundary conditions. In an $L \times L$ square lattice, one contracts over all the vertical bonds between the first and second row of spins. This causes the horizontal edges of these rows to merge to edges of range χ^2. Now, a PEPS state of size $(L-1) \times L$, consisting only of bonds with range χ is sought that approximates this state in an optimal way. The tensors of the $L-2$ upper rows of the new state stay the same as before, the tensors of the now lowest row are optimized, by a horizontal DMRG-style sweeping, to minimize the distance between the two states. Then, this row of newly calculated tensors is again contracted with the tensors above and this is iterated until the whole network is contracted to just a single row. The advantage is that due to the DMRG sweeps, a much better approximation of the true expectation value is achieved. The disadvantage is that the necessity to start at an edge seems to preclude the treatment of periodic-boundary conditions. This is a rather serious impediment as it renders extrapolations to the thermodynamic limit much harder, despite the otherwise good results of the method in open-boundary cases.

9.4 Tensor networks and area laws

The scaling of block-wise entanglement with block size is a hot topic of current research. The starting point of this line of research lies in a rather different realm of physics: in black holes, which are objects of maximum entropy in general relativity, the entropy does not scale with the volume of the black hole, but with the surface area of its event horizon. This was first conjectured in a seminal article by Bekenstein [Bek73], and later proved more rigorously. Such a scaling of entropy-like quantities is, however, not constrained to such extreme cases, as was seen considerably later.

For a quantum system in a pure state with a bipartition into sub-systems A and B, the blockwise entanglement with respect to this bipartition is defined as $S_{A:B} = \operatorname{tr} \rho_A \log_2 \rho_A = \operatorname{tr} \rho_B \log_2 \rho_B$, where ρ_A (ρ_B) is the reduced density matrix of the system state obtained when tracing over the degrees of freedom of sub-system B (A). Remarkably, in the ground states of several very general classes of quantum systems, this quantity is not proportional to the volume of one of the sub-systems but to the area of the boundary of the bipartition. Such an area law was first proved for harmonic oscillators in a chain [AEPW02] or on a lattice [PEDC05], then shown to be true as

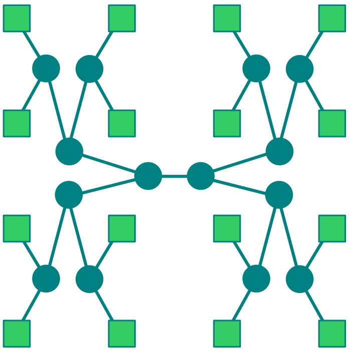


Figure 9.5: A natural way to connect spins on a 4 × 4 lattice to a degree-3 Cayley tree.

well for a large class of non-critical bosonic system in arbitrary dimension [CEPD06]. For fermions, a logarithmic correction occurs [CEP07, Wol06]. In critical systems, entanglement may grow faster —by a logarithmic factor— than the boundary area of the block (studied for chains in [Kor04, KM05] and for higher dimensions in [BCS06]). A proof of an area law at finite temperature has recently been found, too [WVHC07].

These results are very important when assessing whether a given class of variational states is suitable to approximate well the ground state of some system. As even for non-critical 2D systems the blockwise entanglement grows linearly with the boundary of the block, it is imperative to require that the variational trial states do the same. The PEPS class fulfills this requirement, and the valence-bond picture described in the previous section (see Fig. 9.4) shows this in an intuitive way: If a bipartition of a 2D lattice is chosen, the number of lattice bonds that the boundary crosses is proportional to its length. We require the blockwise entanglement to scale linearly in the length of the boundary, too, and, in fact, this can be done: The maximally entangled auxiliary state associated with each crossed bond contributes an amount of up to $\log_2 \chi$ to the entanglement, and these contributions add up. Hence, the PEPS states are well suited to approximate ground states of all systems for which the area law holds [VC04a] and even other systems than can be reduced to such cases [VWPC06].

We may now attempt to use tree tensor states to describe systems on 2D lattices. A natural way to connect the spins on a 2D lattice to a degree-3 Cayley tree is shown in Fig. 9.5. A bipartition of the lattice along the vertical median divides the tree into two parts connected by just one edge. It is easy to see that, for any system size $L \times L$ and any way of building the Cayley tree, there is always a bipartition of boundary length at least L that cuts the lattice such that only a single edge connects the two parts. The blockwise entanglement of the tensor tree state with respect to such a bipartition is bounded by $\log_2 \chi$, for reasons analogous to those discussed in the preceding section. This bound does not grow with L while the blockwise entanglement of most ground states is expected to do so. In order to keep up, χ has to be increased exponentially with L! It is thus unlikely that tensor tree states alone are a viable variational class for two or higher dimensions.

A way out of this dilemma may be to add additional features to the class of states that allow for stronger entanglement. As we shall see in Sec. 9.6 it is possible to let weighted phase gates of the same kind as discussed in Ch. 8 act on all pairs of spins without losing the ability to efficiently calculate expectation values of observables. But first we discuss how to optimize the tensors of a tensor tree state (without weighted phase gates) such that the energy is minimized.

9.5 Optimizing tensor tree states

We now want to use tensor tree states as test states for a variational method, i. e., for a given system Hamiltonian H and a tensor network T describing a state $|\Psi(T)\rangle$, we want to find those values for the tensor entries that minimize the energy

$$\frac{\langle \Psi(T) \,|\, H \,|\, \Psi(T) \rangle}{\langle \Psi(T) \,|\, \Psi(T) \rangle}. \tag{9.4}$$

Note first that $|\Psi(T)\rangle$ is a valid state for any values of the tensor entries unless certain linear dependencies (to be discussed later) render the state to be the zero vector. However, $|\Psi(T)\rangle$ cannot be expected to be normalized.

As the general dependency of the energy on the individual tensor entries may be complicated, we may not expect that finding the global minimum is straight-forward, efficient or possible at all. Fortunately, it turns out, that if one holds all tensor values fixed except for those in one tensor, the map of these values to the energy takes the form of a so-called generalized Rayleigh quotient, which can be globally minimized by solving a so-called generalized eigenvalue problem (see e. g. [NW99]).

To this end, let us pick out one tensor, $T^{[a]}$ and see how the expectation value of a product observable O of form (9.2) depends on the entries of $T^{[a]}$ if all other tensors are considered fixed. Let us write b_1, b_2, b_3 for the three neighbors of the tensor vertex a, and $\chi_i = \chi(a, b_i)$ $(i = 1, 2, 3)$ for the edge ranges of the three edges. Generalizing Eq. (9.3), we see that each tensor is associated with three double sums over its three indices. The tensor appears twice, once with and once without complex conjugation, and is multiplied with factors that only depend on the other tensors and the observable and are hence constant as we consider the other tensors fixed. We assemble these constants to a tensor $\tilde{O}$ and get

$$\langle \Psi(T) \,|\, O \,|\, \Psi(T) \rangle = \sum_{\alpha_1',\alpha_1=0}^{\chi_1-1} \sum_{\alpha_2',\alpha_2=0}^{\chi_2-1} \sum_{\alpha_3',\alpha_3=0}^{\chi_3-1} T^{[a]*}_{\alpha_1'\alpha_2'\alpha_3'} T^{[a]}_{\alpha_1\alpha_2\alpha_3} \tilde{O}_{\alpha_1'\alpha_2'\alpha_3'\alpha_1\alpha_2\alpha_3} \tag{9.5}$$

The elements of the tensor $\tilde{O}$ can be calculated explicitly (Keep in mind here, that for each active tensor a, we get a different tensor $\tilde{O}$.): Let us denote by $\tilde{T}(\alpha_1, \alpha_2, \alpha_3)$ the tensor network T with the following alteration: In $\tilde{T}(\alpha_1, \alpha_2, \alpha_3)$, all tensors $T^{[b]}$ are the same as in T for $b \neq a$, but the tensor $T^{[a]}$ is replaced by a tensor that has zeroes in all entries except for the entry $\tilde{T}^{[a]}_{\alpha_1,\alpha_2,\alpha_3}$, which is 1. With this definition, it is easy to see, that

$$\tilde{O}_{\alpha_1'\alpha_2'\alpha_3'\alpha_1\alpha_2\alpha_3} = \left\langle \Psi\left(\tilde{T}(\alpha_1'\alpha_2'\alpha_3')\right) \,\middle|\, O \,\middle|\, \Psi\left(\tilde{T}(\alpha_1\alpha_2\alpha_3)\right) \right\rangle.$$

Hence, we can use the algorithm described in Sec. 9.2 to compute the tensor $\tilde{O}$. (Note that the facts just presented for the product observable O also hold for any sum of product observables, such as the system Hamiltonian H.) We only have to use the contraction algorithm for each term in H in order to obtain a tensor $\tilde{E}$, that plays, for H, the same role as $\tilde{O}$ does for O in Eq. (9.5).

We now come back to the task of minimizing the expression (9.4). We assemble the index triple $(\alpha_1, \alpha_2, \alpha_3)$ that occurred in Eq. (9.5) to a single index

$$\alpha = \alpha_1 + \chi_1\alpha_2 + \chi_1\chi_2\alpha_3.$$

Indexed in this manner, the three-index tensor $T^{[a]}$ becomes a one-index tensor $\mathbf{T}^{[a]}$, i. e., a vector. Likewise, we use the same indexing to regard the 6-index tensor $\tilde{O}$ as a matrix, and write henceforth $\tilde{O}_{\alpha'\alpha}$ for $\tilde{O}_{\alpha_1'\alpha_2'\alpha_3'\alpha_1\alpha_2\alpha_3}$. Eq. (9.5) then becomes

$$\langle \Psi(T) \,|\, O \,|\, \Psi(T) \rangle = \sum_{\alpha',\alpha=0}^{\chi_1\chi_2\chi_3-1} T^{[a]*}_{\alpha'} \tilde{O}_{\alpha'\alpha} T^{[a]}_{\alpha} = \mathbf{T}^{[a]\dagger} \tilde{O} \mathbf{T}^{[a]}.$$

Next, we introduce the tensor $\tilde{E}$ that corresponds to the Hamiltonian in the same way as $\tilde{O}$ corresponds to O and will be read as a matrix. We also define a tensor or matrix $\tilde{N}$ that corresponds in the same way to the identity "observable" $\mathbb{1}$. This allows us to rewrite expression (9.4) as

$$\frac{\mathbf{T}^{[a]\dagger} \tilde{E} \mathbf{T}^{[a]}}{\mathbf{T}^{[a]\dagger} \tilde{N} \mathbf{T}^{[a]}} \tag{9.6}$$

This expression is the quotient of two quadratic forms and such an expression is known as generalized Rayleigh quotient. It can be minimized globally with respect to the entries in $\mathbf{T}^{[a]}$ by solving the generalized eigenvalue problem for the matrix pencil $\tilde{E} - \lambda\tilde{N}$, i. e.

$$\tilde{E}\mathbf{T} = \lambda\tilde{N}\mathbf{T}. \tag{9.7}$$

The generalized eigenvector $\mathbf{T}_0$ corresponding to the smallest generalized eigenvalue λ_0 minimizes the energy and λ_0 is this minimum energy. Hence, by solving the generalized eigenvalue problem and then replacing $T^{[a]}$ with $\mathbf{T}_0$ (now read again as 3-index tensor), we have found the global minimum of the energy as a function of the elements of $T^{[a]}$. At this point, it is important to remark that even though we have found a global minimum here, it is merely a global minimum for the restricted energy function that only allows the entries in one tensor to be varied. There is a priori no reason to assume that performing these steps for each tensor will lead to the true global minimum of the optimization problem (9.5). However, numerical experiments show that iterating this procedure several times over all tensors gives very good results and seems to converge to the true global minimum in most cases, although it can be proved that this does not always hold (by adapting the proof of [Eis06]).

We need to solve the generalized eigenvalue problem (9.7). Algorithms for this well-studied problem have been implemented, most notably in all implementations of LAPACK [ABB⁺99] and its supersets. As both $\tilde{E}$ and $\tilde{N}$ are Hermitean, we may use a routine which takes advantage of this such as LAPACK's routine ZHEGV, which performs a Cholesky factorization[6] to reduce the problem to an ordinary eigenvalue problem, which is then solved by Lanczos iteration. (For non-Hermitean matrices, LAPACK offers the routine ZGEGV, which implements the QZ algorithm [MS73].)

9.5.1 Dealing with linear dependencies

When optimizing the tensor $T^{[a]}$, it is helpful to regard the tensor tree state as a sum of the form

$$|\Psi(T)\rangle = \sum_{\alpha_1=0}^{\chi_1} \sum_{\alpha_2=0}^{\chi_2} \sum_{\alpha_3=0}^{\chi_3} T^{[a]}_{\alpha_1\alpha_2\alpha_3} |\alpha_1\rangle_1 |\alpha_2\rangle_2 |\alpha_3\rangle_3 \,, \tag{9.8}$$

where the ket $|\alpha\rangle_j$ denotes the state that lives on the spins that are connected with tensor vertex a via its edge j and that results from contraction along all the edges connected to the tensor vertex via edge j if the index of this edge is fixed to the value α. In this notation, the meaning of the elements of the matrices $\tilde{N}$ and $\tilde{E}$ in the generalized Rayleigh coefficient may then be understood as

$$\begin{aligned}
\tilde{N}_{\alpha'_1\alpha'_2\alpha'_3,\alpha_1\alpha_2\alpha_3} &= {}_3\langle\alpha'_3|\,{}_2\langle\alpha'_2|\,{}_1\langle\alpha'_1 \mid \alpha_1\rangle_1 |\alpha_2\rangle_2 |\alpha_3\rangle_3 \\
\tilde{E}_{\alpha'_1\alpha'_2\alpha'_3,\alpha_1\alpha_2\alpha_3} &= {}_3\langle\alpha'_3|\,{}_2\langle\alpha'_2|\,{}_1\langle\alpha'_1 \mid H \mid \alpha_1\rangle_1 |\alpha_2\rangle_2 |\alpha_3\rangle_3 \,.
\end{aligned}$$

If now one of the sets $\left\{|\alpha\rangle_j\right\}_{\alpha=0}^{\chi_j-1}$ $(j = 1, 2, 3)$ is linearly dependent, the matrix $\tilde{N}$ becomes singular and some of its eigenvalues vanish. Then, $\tilde{N}$ ceases

---

[6] A good textbook on numerical linear algebra that explains these terms is [TB97]

to have a Cholesky factorization, and the Cholesky–Lanczos solver cannot be used any longer. We have tried to use the QZ algorithm [MS73] instead, which is also implemented in LAPACK. However, it is numerically unstable and seems to drift ever deeper into the region of singular $\tilde{N}$ matrices. This does not seem to be the fault of the QZ algorithm, but rather because generalized eigenvalue problems with singular or near-singular matrices are inherently highly sensitive to small perturbations [Ste72].

Hence, we try to "cut off" the singular part. To this end, we first diagonalize $\tilde{N}$,

$$\tilde{N} = UDU^\dagger,$$

where the diagonal matrix D contains the eigenvalues in descending order, so that all vanishing eigenvalues are found at the bottom. (There cannot be any truly negative eigenvalues, and negative or positive values very close to zero shall be deemed numerically zero, and the imprecision only due to rounding errors.) Let s be the number of zero eigenvalues, and D' be the $(\xi - s) \times (\xi - s)$ diagonal matrix of the non-vanishing eigenvalues (writing $\xi = \chi_1\chi_2\chi_3$ for the size of $\tilde{N}$ and $\tilde{E}$). If we cut away the s right-most columns of U, yielding the rectangular $\xi \times (\xi - s)$ matrix R, we have

$$D' = R^\dagger \tilde{N} R.$$

We transform $\tilde{E}$ similarly to the $(\xi - s) \times (\xi - s)$ matrix

$$E' = R^\dagger \tilde{E} R.$$

We have lost no information by reducing the size of $\tilde{E}$. To see this, note that the original tensor $T^{[a]}$ can as well afford to lose this information. Let us rewrite the $\chi_1 \times \chi_2 \times \chi_3$ tensor $T^{[a]}$ as a vector $\mathbf{T}_a$ of length $\xi = \chi_1\chi_2\chi_3$, transform it to length $\xi - s$ and back to length ξ by the operation $\tilde{\mathbf{T}}_a = RR^\dagger \mathbf{T}_a$. The state $|\Psi(t)\rangle$ as written in Eq. (9.8) does not change if $T^{[a]}$ is replaced by $\tilde{\mathbf{T}}_a$ (written as a tensor again), as can be seen from a straight-forward calculation.

As D' is now manifestly positive definite, the generalized eigenvalue problem $E'\mathbf{T}' = \lambda D'\mathbf{T}'$ can be solved easily. As inversion of D' is trivial (just replace the diagonal entries by its reciprocals), we just have to solve an ordinary eigenvalue problem $D'^{-1}E'\mathbf{T}' = \lambda \mathbf{T}'$. The eigenvector $\mathbf{T}'_0$ corresponding to the lowest eigenvalue λ_0 is now a vector of length $\xi - s$, but can be transformed to length ξ using R,

$$\mathbf{T}_0 = R\mathbf{T}'_0.$$

The vector $\mathbf{T}_0$, written again as rank-3 tensor of size $\chi_1 \times \chi_2 \times \chi_3$ is used to overwrite the old tensor T.

9.6 Combining tensor trees and weighted graphs

At the end of 9.4, we noticed that we need to enhance the class of tensor tree states in order to make them suitable for 2D lattices. Here, we show how observables can still be calculated efficiently if we augment the tensor tree states by letting weighted phase gates act on all pairs of spins, because such an augmented state is capable of showing the required amount of entanglement (see Sec. 8.3.3). The fact, that such a combination is possible without losing the ability to efficiently evaluate observables has first been noticed by F. Verstraete and W. Dür [pers. comm.] and independently by J. Eisert and M. Plenio [pers. comm.]. The algorithm that I present here is simpler and more efficient than the original ansatz of Dür and Verstraete.

As in Chapter 8, we define the generalized phase gate, acting on two d-level systems, as

$$W_\Phi = \sum_{s_1,s_2=0}^{d-1} |s_1 s_2\rangle \langle s_1 s_2| \, e^{i\Phi_{s_1 s_2}}, \tag{9.9}$$

where the $d \times d$ matrix Φ contains the phases. We shall require, as in Ch. 8, that $\Phi = \Phi^T$ and that $\Phi_{0,s} = \Phi_{s,0} = 0$ ($s = 0, \dots, d-1$). As before, Φ is a four-index tensor $\Phi^{ab}_{s_1 s_2}$ where the lower two indices specify the levels s_1, s_2 (cf. Eq. (9.9)) and the upper two the spin indices $a, b \in \{1, \dots, N\}$. With this, we define the operator $\mathcal{W}_\Phi$, which applies all phase gates,

$$\mathcal{W}_\Phi = \prod_{\substack{a,b \\ a<b}} W^{(a,b)}_{\Phi^{ab}}.$$

We consider states that result from applying $\mathcal{W}_\Phi$ onto a tensor tree state $|\Psi(T)\rangle$ with tensor network T. We shall see that for a state $\mathcal{W}_\Phi |\Psi(T)\rangle$ it is possible to efficiently calculate the expectation value of any observable that can be written as sum of K terms, each of which is a tensor product of local operators of which only a small number k is not the identity. The time to calculate such an expectation value scales obviously linearly in K and, as we shall see, exponential in k.

Often, the observable of interest is the Hamiltonian H, and for many systems H is a sum of bi-local terms,

$$H = \sum_{(a,b)\in\mathcal{B}} H^{(a,b)}_{ab},$$

where $\mathcal{B}$ is the set of bonds. Bonds are pairs of qubits acted on jointly by a term in H. We restrict ourselves to this case to keep notation simple. The energy is

$$E = \sum_{(a,b)\in\mathcal{B}} \langle \Psi(T)| \, \mathcal{W}^\dagger_\Phi H^{(ab)}_{ab} \mathcal{W}_\Phi \, |\Psi(T)\rangle = \sum_{(a,b)\in\mathcal{B}} \operatorname{tr} H_{ab} \rho_{ab},$$

and we now discuss how to calculate the reduced density matrices

$$\begin{aligned}\rho_{ab} &= \operatorname{tr}_{V\setminus\{a,b\}} \mathcal{W}_\Phi \left|\Psi(T)\right\rangle \left\langle\Psi(T)\right| \mathcal{W}_\Phi^\dagger \\ &= W_{\Phi^{ab}} \underbrace{\left(\operatorname{tr}_{V\setminus\{a,b\}} \mathcal{W}_{ab} \left|\Psi(T)\right\rangle \left\langle\Psi(T)\right| \mathcal{W}_{ab}^\dagger\right)}_{:=\tilde{\rho}_{ab}} W_{\Phi^{ab}}^\dagger . \\ & \qquad\qquad (V := \{1,\dots,N\}) \qquad (9.10)\end{aligned}$$

Here, we have pulled $W_{\Phi^{ab}}^{(ab)}$ out of the trace, and also taken advantage of the cyclicity of the trace, which allows us to omit from $\mathcal{W}_\Phi$ all those terms which do not act on a or b. We denote the operators remaining in the trace with

$$\mathcal{W}_{ab} := \prod_{c\in V\setminus\{a,b\}} W_{\Phi^{ac}}^{(ac)} W_{\Phi^{bc}}^{(bc)} .$$

Finally, we have to calculate $\tilde{\rho}_{ab}$, the underbraced part of Eq. (9.10). To this end, we introduce the local diagonal operator

$$V_{\boldsymbol{\Phi}} := \sum_{s=0}^{d-1} \left|s\right\rangle \left\langle s\right| e^{i\Phi_s}$$

and observe that

$$W_\Phi = \sum_{s=0}^{d-1} \left|s\right\rangle \left\langle s\right| \otimes V_{\boldsymbol{\Phi}_{\cdot,s}} , \qquad (9.11)$$

where $\boldsymbol{\Phi}_{\cdot,s}$ denotes the s-th column (starting to count from 0) of the matrix Φ.

For the next step, note that the matrix elements $\left\langle s_a' s_b' \,\middle|\, \tilde{\rho}_{ab} \,\middle|\, s_a s_b\right\rangle$ of $\tilde{\rho}_{ab}$ can trivially be written as follows:

$$\begin{aligned}&\left\langle s_a' s_b' \,\middle|\, \tilde{\rho}_{ab} \,\middle|\, s_a s_b\right\rangle \\ &\qquad = \left\langle\Psi(T)\right| \left[\mathcal{W}_{ab} \left(\left|s_a\right\rangle \left\langle s_a'\right|\right)^{(a)} \left(\left|s_b\right\rangle \left\langle s_b'\right|\right)^{(b)} \mathcal{W}_{ab}^\dagger\right] \left|\Psi(T)\right\rangle \qquad (9.12)\end{aligned}$$

If the expression in square brackets were a product observable we could calculate the matrix element using the contraction algorithm of Sec. 9.2 on the tensor tree state $\left|\Psi(T)\right\rangle$. The trick is now to note that despite its appearance, the expression in square brackets in fact *is* a tensor product of local observables. To see this, we expand $\mathcal{W}_{ab}$ using Eq. (9.11),

$$\mathcal{W}_{ab} = \prod_{c\in V\setminus\{a,b\}} \left[\sum_{s_a=0}^{d-1} \left(\left|s_a\right\rangle \left\langle s_a\right|\right)^{(a)} V^{(c)}_{\boldsymbol{\Phi}^{ac}_{\cdot,s_a}}\right] \left[\sum_{s_b=0}^{d-1} \left(\left|s_b\right\rangle \left\langle s_b\right|\right)^{(b)} V^{(c)}_{\boldsymbol{\Phi}^{bc}_{\cdot,s_b}}\right]$$

and insert the expansion into Eq. (9.12). After cancelling all but one of the many operators $\left(\left|s_a\right\rangle \left\langle s_a\right|\right)^{(a)}$ and $\left(\left|s_b\right\rangle \left\langle s_b\right|\right)^{(b)}$, we use the "linearity" of $V_{\boldsymbol{\Phi}}$

(i. e. $V_{\mathbf{\Phi}_1} V_{\mathbf{\Phi}_2} = V_{\mathbf{\Phi}_1 + \mathbf{\Phi}_2}$ and $V_{\mathbf{\Phi}}^\dagger = V_{-\mathbf{\Phi}}$), and are left with

$$\begin{aligned} &\left\langle s'_a s'_b \,\middle|\, \tilde{\rho}_{ab} \,\middle|\, s_a s_b \right\rangle \\ &\qquad = \langle \Psi(T) | \Bigg[\left(|s'_a\rangle \langle s_a| \right)^{(a)} \left(|s'_b\rangle \langle s_b| \right)^{(b)} \times \\ &\qquad\qquad \times \prod_{c \in V \setminus \{a,b\}} V^{(c)}_{\mathbf{\Phi}^{ac}_{\cdot,s_a} - \mathbf{\Phi}^{ac}_{\cdot,s'_a} + \mathbf{\Phi}^{bc}_{\cdot,s_b} - \mathbf{\Phi}^{bc}_{\cdot,s'_b}} \Bigg] |\Psi(T)\rangle \,, \end{aligned} \tag{9.13}$$

which is manifestly a product observable.

Note that this form enables us to use the optimization scheme of Sec. 9.5 in order to update the tensors T to yield a state $\mathcal{W}_\Phi |\Psi(T)\rangle$ with minimal energy. When using this scheme, the phases Φ stay fixed while the tensors in T are optimized. In order to optimize the phases, we may keep the tensors fixed and use a gradient-based minimization as in Sec. 8.6.

9.6.1 Implementation

When carrying out a variational computation with tensor tree states, the dominant computational effort is the calculation of the matrices $\tilde{E}$ and $\tilde{N}$ in the Rayleigh quotient (9.6). For each element, a contraction of the kind described in Sec. 9.2 has to be carried out, and hence this operation has to be highly optimized. I have programmed the contraction in C++ and then used SWIG [B+] to generate bindings that allow to use the C++ classes representing the tensor and leaf vertices from Python. This approach follows again the strategy outlined in Sec. 8.A.2 where we coded only the inner loops in a high-performance language such as C++ and use a modern dynamically-typed language for the rest, which allows for more rapid development for the rest. For the latter we have again chosen Python. The fixed interface between the C++ and the Python part allowed all researchers involved in the tree tensor project to work on the Python parts without any need to learn about the internals of the C++ core; in fact, without any need to learn the C++ programming language at all.

The strategy turned out to be fruitful, and we have produced code to test a wide variety of settings: Tensor-tree states without and with phase gates, and, for comparison with DMRG, matrix-product states in tensor-tree form (as in Fig. 9.2), also routines to describe different lattices in one and two dimensions and different Hamiltonians, including the XY, XXZ, Bose-Hubbard, and fermionic Hubbard models. Using the University's high-performance compute clusters, we are, at the time of this writing, testing the performance of the scheme for various choices of χ, various forms of the tree and various symmetry restrictions of the phases.

In order to get descent computing times, we had to carefully think about the control flow in the computation: For instance, it turned out to be absolutely crucial to cache the results for sub-trees when going from one term of the Hamiltonian to the next in the calculation of $\tilde{E}$, and to even retain intermediate results when going from one active tensor to the next. For the optimization of the phases, a gradient has to be obtained (similar to Sec. 8.B) in order to use a gradient-based minimizer such as L-BFGS [BLN95]. Even then, the computational effort is still too large unless one attempts further simplifications such as to only approximate the tree contraction during the phase optimization rounds.

9.6.2 Preliminary results and conclusions

At the time of this writing, we are still running calculations in order to see for which systems the tensor trees, possibly with the weighted-phase gate extension, are viable and for which not. Unfortunately, first results are not too encouraging. We have put much effort in finding optimal algorithms, especially with respect to caching (i. e., storage of intermediate results for later re-use) which brought down the time complexity considerably. We also used advanced programming techniques to make sure that the code makes optimal use of the machine's capabilities. Nevertheless, convergence is clearly too slow and too weak.

Our efforts should be seen in the context of their competition with other techniques. As we have seen in the overview of Ch. 6, there is a large variety of techniques for treating spin systems to chose from. For one-dimensional systems, DMRG is clearly the best choice to find ground states. It converges quite fast and reaches an accuracy that leaves little to be desired, and thus there is hardly any room left for further improvements. This was the reason why we intended from the beginning to focus on two-dimensional systems. Our attempts to use weighted-graph states for this, as described in Chs. 7 and 8, have been of rather limited success: The system sizes that we could handle were very modest, and the accuracy of the approximate state was insufficient to see global properties such as momentum distributions. As we have seen in Sec. 8.7 this is due to the highly non-linear dependence of observable expectation values on the parameters that lead, as we have found, to a huge amount of local minima. A generic optimization algorithm cannot be expected to find a sufficiently good minima among these many possibilities.

DMRG variants might be expected to fare better in two dimensions. Since the first publication [VC04a], the authors of the PEPS method have improved their implementation and recently presented results for a Bose-Hubbard system [MVC07], which are, however, still limited to 11×11 sites. This cannot match with path-integral Monte Carlo in terms of system size and accuracy. (Ref. [WATB04] treats 50×50 sites and does not need to resort to the hardcore approximation.) On the other hand, the PEPS ansatz shows significant promise for time evolution (as demonstrated by an example calculation in [MVC07]) and might there be ahead of quantum Monte Carlo. According to the authors of the PEPS papers [I. Cirac, pers. comm.], the main obstacles for a more profitable use of the methods are (i) the inability to cope with periodic boundary conditions (which severely limits the possibility to extrapolate to the thermodynamic limit) and (ii) the adverse scaling with the dimension of the spins. Hence our hope that tensor trees might overcome these problems, because they have a benign scaling with spin dimension (see Sec. 9.2). Of course, tensor tree states cannot be expected to cope well with the entanglement in two dimensions due to the area law (see Sec. 9.4), while the entanglement properties of weighted graph states seems well suited for this (see Sec. 8.3.3), no matter whether boundary conditions are open or periodic. The ansatz to combine these two techniques (see Sec. 9.6) seemed thus a natural choice.

In this combination, we depend crucially on the phase gates to provide the necessary entanglement, and it does not come as a complete surprise that we experience problems similar to those described in Ch. 8. When optimizing the phases the convergence is very slow and weak. Often, the energy would hardly change for several hour of calculation time only to then suddenly and unexpectedly start to drop again, as if the minimization had to traverse a long, shallow "trough" of the potential landscape to reach a "slide" at its end. This not only costs time —we need several days for just a 6×6 system of Ising spins—, it also keeps the state at rather high energies. We had the hope that alternating the tensor and the phase optimization may allow to use the known good convergence of the former to boost the latter and push it out of shallow troughs and local minima. While it is not entirely implausible to expect this, our calculations so far do not show such synergy. Rather, the results we got so far are significantly behind any of the other mentioned techniques in terms of precision as well as speed.

As we are still running calculations and have not yet entirely run out of possibilities to tweak the code or improve the algorithms it is to early to give a final verdict. Nevertheless, a claim of success can definitely not be made yet. It is likely that the conclusion of Sec. 8.7 will have to be upheld, namely that a variational method based on phase gates needs an optimization algorithm specifically tailored to this problem. This can only be achieved if deep insights are gained in how the potential landscape is shaped by the specific parametrization of the manifold of our class of variational states.

Appendix

Appendix A

Software used

A large variety of software was used in the work leading to this thesis. Many of these programs are open source software, to a good part created in the course of academic research. Hence is is proper to list and acknowledge at least the most important of these programs. Giving a full list is probably nearly impossible due to the vast amount of intertwined programs in modern computer systems.

All work was done using the GNU/Linux operating system, making extensive use of the GNU Tools, especially the Bourne Again Shell (bash) and the other GNU Core Utils (including the GNU Text Utils). (See [FSF] for more information on the GNU Project). NEdit was used as text editor.

The Mozilla suite and its derivatives was crucial for information access.

Programming was done with the GNU Compiler Collection (GCC, [GCC]), most importantly with the GNU C++ compiler, but also with Intel's C++ compiler. Perl and especially Python [R$^+$] were used extensively, as well as SWIG (the "simple wrapper and interface generator" [B$^+$]). The GNU Debugger (gdb) was used for debugging, along with Python's debugging facilities. For symbolic calculations, Wolfram's Mathematica was employed.

Several numerics libraries were used, including the GNU Scientific Library (GSL, [GTJ$^+$03]), the NumPy system [Num], and the Intel Math Kernel Library (MKL) as highly optimized implementation of BLAS [BDD$^+$02] and LAPACK [ABB$^+$99], next to ATLAS [WPD01] as alternative implementation.

For presentation of the results, the TEX/LATEX distribution teTeX was relied on, and for plots and illustrations, Gnuplot, Grace, and OpenOffice.org were used.

For completeness of the biography, I would also like to mention those computing references and manuals that I relied on most heavily: For Python, the books [LA04] and especially [Mar03], for C++ the classic textbook [Str97] and the standard library reference [Jos99], for LATEX the works [MG04, Grä97, Pak02].

Appendix B

Bibliography

[AAC$^+$07] A. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhenikov, A. Läuchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel (ALPS collaboration). *The ALPS project release 1.3: open source software for strongly correlated systems*. J. Magn. Mag. Mat. **310** (2007), 1187.

[AB97] D. Aharonov, M. Ben-Or. *Fault-tolerant quantum computation with constant error*. In: *STOC '97: Proc. 29th annual ACM symp. on theory of computing*, pp. 176–188. ACM Press, New York, NY, USA (1997). [DOI link]. ArXiv: quant-ph/9611025.

[AB99] D. Aharonov, M. Ben-Or. *Fault-tolerant quantum computation with constant error rate*. ArXiv: quant-ph/9906129 (1999).

[AB06] S. Anders, H. J. Briegel. *Fast simulation of stabilizer circuits using a graph-state representation*. Phys. Rev. A **73** (2006), 022334.

[ABB$^+$99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Cruz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen. *LAPACK Users' Guide*. 3rd edn. SIAM, Philadelphia (1999). `http://www.netlib.org/lapack/lug/lapack_lug.html`.

[ABD07] S. Anders, H. J. Briegel, W. Dür. *A variational method based on weighted graph states*. ArXiv: 0706.0423v2 [quant-ph] (2007). To appear in New J. Phys.

[ADB05] H. Aschauer, W. Dür, H.-J. Briegel. *Multiparticle entanglement purification for two-colorable graph states*. Phys. Rev. A **71** (2005), 012319.

[ADG$^+$05] F. Alet, P. Dayal, A. Grzesik, A. Honecker, M. Körner, A. Läuchli, S. R. Manmana, I. P. McCulloch, F. Michel, R. M. Noack, G. Schmid, U. Schollwöck, F. Stöckli, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel (ALPS collaboration). *The ALPS Project: Open source software for strongly correlated systems*. J. Phys. Soc. Jpn. (Suppl.) **74** (2005), 30. ArXiv: cond-mat/0410407.

[AEPW02] K. Audenaert, J. Eisert, M. B. Plenio, R. F. Werner. *Entanglement properties of the harmonic chain*. Phys. Rev. A **66** (2002), 042327.

[AFOV07] L. Amico, R. Fazio, A. Osterloh, V. Vedral. *Entanglement in many-body systems*. ArXiv: quant-ph/0703044 (2007).

[AG04] S. Aaronson, D. Gottesman. *Improved simulation of stabilizer circuits*. Phys. Rev. A **70** (2004), 052328.

[AGP07] P. Aliferis, D. Gottesman, J. Preskill. *Accuracy threshold for postselected quantum computation*. ArXiv: quant-ph/0703264 (2007).

[AGR81] A. Aspect, P. Grangier, G. Roger. *Experimental tests of realistic local theories via Bell's theorem*. Phys. Rev. Lett. **47** (1981), 460.

[Aha99] D. Aharonov. *Noisy quantum computation*. Ph. D. thesis, Hebrew University, Jerusalem (1999). `http://www.cs.huji.ac.il/~doria/thesis.ps`.

[AK] S. Aaronson, G. Kuperberg. *Complexity Zoo*. `http://qwiki.caltech.edu/wiki/Complexity_Zoo`.

[Aki96] H. Akima. *Algorithm 761: scattered-data surface fitting that has the accuracy of a cubic polynomial*. ACM Trans. Math. Softw. **22** (1996), 362.

[AKLT87] I. Affleck, T. Kennedy, E. H. Lieb, H. Tasaki. *Rigorous results on valence-bond ground states in antiferromagnets*. Phys. Rev. Lett. **59** (1987), 799.

[And87] P. W. Anderson. *The resonating valence bond state in* La_2CuO_4 *and superconductivity*. Science **235** (1987), 1196.

[APD$^+$06] S. Anders, M. B. Plenio, W. Dür, F. Verstraete, H.-J. Briegel. *Ground state approximation for strongly interacting systems in arbitrary dimension*. Phys. Rev. Lett. **97** (2006), 107206.

[Aue94] A. Auerbach. *Interacting electrons and quantum magnetism*. Springer, New York etc. (1994).

[B$^+$] D. M. Beazley, et al. *Simple Wrapper and Interface Generator (SWIG)*. `http://www.swig.org`.

[BA57] D. Bohm, Y. Aharonov. *Discussion of experimental proof for the paradox of Einstein, Rosen, and Podolsky*. Phys. Rev. **108** (1957), 1070.

[Bak61] G. A. Baker. *Application of the Padé approximant method to the investigation of some magnetic properties of the Ising model*. Phys. Rev. **124** (1961), 768.

[Bak90] G. A. Baker. *Quantitative theory of critical phenomena*. Academic Press, San Diego (1990).

[BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani. *Strengths and weaknesses of quantum computing*. SIAM J. Comp. **26** (1997), 1510.

[BBP+96] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, W. K. Wootters. *Purification of noisy entanglement and faithful teleportation via noisy channels*. Phys. Rev. Lett. **76** (1996), 722.

[BCG05] O. Bröker, O. Chinellato, R. Geus. *Using Python for large scale linear algebra applications*. Future Gener. Comput. Syst. **21** (2005), 969.

[BCJD99] G. K. Brennen, C. M. Caves, P. S. Jessen, I. H. Deutsch. *Quantum logic gates in optical lattices*. Phys. Rev. Lett. **82** (1999), 1060.

[BCS06] T. Barthel, M.-C. Chung, U. Schollwöck. *Entanglement scaling in critical two-dimensional fermionic and bosonic systems*. Phys. Rev. A **74** (2006), 022329.

[BDD+02] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, R. C. Whaley. *An updated set of basic linear algebra subprograms (BLAS)*. ACM Trans. Math. Softw. **28** (2002), 135.

[BDL07] S. Bravyi, D. DiVincenzo, D. Loss. *Polynomial-time algorithm for simulation of weakly interacting quantum spin systems*. ArXiv: 0707.1894 [quant-ph] (2007).

[BDSW96] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, W. K. Wootters. *Mixed-state entanglement and quantum error correction*. Phys. Rev. A **54** (1996), 3824.

[Bek73] J. D. Bekenstein. *Black Holes and Entropy*. Phys. Rev. D **7** (1973), 2333.

[Bel64] J. S. Bell. *On the Einstein-Podolsky-Rosen paradox*. Physics **1** (1964), 195. Reprinted in: J. S. Bell. *Speakable and unspeakable in quantum mechanics*. Cambridge University Press, Cambridge (1987).

[BEZ00] D. Bouwmeester, A. K. Ekert, A. Zeilinger (Eds.). *The physics of quantum information*. Springer (2000).

[BKN00] H. Barnum, E. Knill, M. A. Nielsen. *On quantum fidelities and channel capacities*. IEEE Trans. Info. Theory **46** (2000), 1317.

[BL07] D. Bruß, G. Leuchs (Eds.). *Lectures on quantum information*. Wiley-VCH, Weinheim (2007).

[BLN95] R. H. Byrd, P. Lu, J. Nocedal. *A limited memory algorithm for bound constrained optimization*. SIAM J. Sci. Stat. Comp. **16** (1995), 1190.

[BM71] E. Barouch, B. M. McCoy. *Statistical mechanics of the XY model. II. Spin-correlation functions*. Phys. Rev. A **3** (1971), 786.

[Boh51] D. Bohm. *Quantum theory*. Prentice-Hall, New York (1951).

[BR01] H.-J. Briegel, R. Raußendorf. *Persistent entanglement in arrays of interacting particles*. Phys. Rev. Lett. **86** (2001), 910.

[Bro07] D. E. Browne. *Efficient classical simulation of the quantum Fourier transform*. New J. Phys. **9** (2007), 146.

[BV93] E. Bernstein, U. Vazirani. *Quantum complexity theory*. In: *Proc. 25th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 11–20. ACM (1993). See also [BV97], which is an extended version of this article.

[BV97] E. Bernstein, U. Vazirani. *Quantum complexity theory*. SIAM J. Comp. **26** (1997), 1411.

[BXG96] R. J. Bursill, T. Xiang, G. A. Gehring. *The density matrix renormalization group for a quantum spin chain at non-zero temperature*. J. Phys.: Cond. Mat. **8** (1996), L583.

[CCK77] D. Ceperley, G. V. Chester, M. H. Kalos. *Monte Carlo simulation of a many-fermion study*. Phys. Rev. B **16** (1977), 3081.

[Cep95] D. M. Ceperley. *Path integrals in the theory of condensed helium*. Rev. Mod. Phys. **67** (1995), 279.

[CEP07] M. Cramer, J. Eisert, M. B. Plenio. *Statistics dependence of the entanglement entropy*. Phys. Rev. Lett. **98** (2007), 220603.

[CEPD06] M. Cramer, J. Eisert, M. B. Plenio, J. Dreissig. *Entanglement-area law for general bosonic harmonic lattice systems*. Phys. Rev. A **73** (2006), 012309.

[CGS02] C. Crépeau, D. Gottesman, A. Smith. *Secure multi-party quantum computation*. In: *Proc. 34th Annual ACM Symp. on Theory of Computing (STOC)*, pp. 643–652. ACM (2002). Also on ArXiv: quant-ph/0206138).

[CHDB05] J. Calsamiglia, L. Hartmann, W. Dür, H.-J. Briegel. *Spin gases: quantum entanglement driven by classical kinematics*. Phys. Rev. Lett. **95** (2005), 180502.

[CHSH69] J. F. Clauser, M. A. Horne, A. Shimony, R. A. Holt. *Proposed experiment to test local hidden-variable theories*. Phys. Rev. Lett. **23** (1969), 880.

[CL04] K. Chen, H.-K. Lo. *Multi-partite quantum cryptographic protocols with noisy GHZ states*. ArXiv: quant-ph/0404133 (2004).

[Cla04] F. Clare. *NATGRID*. Part of the *NCAR Graphics* software library, `http://ngwww.ucar.edu/ngdoc/ng/ngmath/natgrid/intro.html` (2004). [based on *nngridr* by D. Watson (1994)].

[CLM04] S. Capponi, A. Läuchli, M. Mambrini. *Numerical contractor renormalization method for quantum spin models*. Phys. Rev. B **70** (2004), 104424.

[CM02] M. A. Cazalilla, J. B. Marston. *Time-dependent density-matrix renormalization group: a systematic method for the study of quantum many-body out-of-equilibrium systems*. Phys. Rev. Lett. **88** (2002), 256403.

[CRT$^+$03] A. Cuccoli, T. Roscilde, V. Tognetti, R. Vaia, P. Verrucchi. *Quantum Monte Carlo study of $S = \frac{1}{2}$ weakly anisotropic antiferromagnets on the square lattice*. Phys. Rev. B **67** (2003), 104414.

[CS96] A. R. Calderbank, P. W. Shor. *Good quantum error-correcting codes exist*. Phys. Rev. A **54** (1996), 1098.

[CTS00] L. Capriotti, A. E. Trumper, S. Sorella. *Ground state properties of the S=1/2 Heisenberg antiferromagnet on the triangular lattice*. J. Appl. Phys. **87** (2000), 7046.

[CZ95] J. I. Cirac, P. Zoller. *Quantum computations with cold trapped ions*. Phys. Rev. Lett. **74** (1995), 4091.

[DA07] D. P. DiVincenzo, P. Aliferis. *Effective fault-tolerant quantum computation with slow measurements*. Phys. Rev. Lett. **98** (2007), 020501.

[DAB03] W. Dür, H. Aschauer, H. J. Briegel. *Multiparticle entanglement purification for graph states*. Phys. Rev. Lett. **91** (2003), 107903.

[DB04] W. Dür, H.-J. Briegel. *Stability of macroscopic entanglement under decoherence*. Phys. Rev. Lett. **92** (2004), 180403.

[DCH04] W. Dür, J. I. Cirac, P. Horodecki. *Nonadditivity of quantum capacity for multiparty communication channels*. Phys. Rev. Lett. **93** (2004), 020503.

[DEJ$^+$96] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, A. Sanpera. *Quantum privacy amplification and the security of quantum cryptography over noisy channels*. Phys. Rev. Lett. **77** (1996), 2818.

[DEO07] C. M. Dawson, J. Eisert, T. J. Osborne. *Unifying variational methods for simulating quantum many-body systems*. ArXiv: 0705.3456 [quant-ph] (2007).

[Deu85] D. Deutsch. *Quantum theory, the Church-Turing principle and the universal quantum computer*. Proc. Roy. Soc. Lon. A **400** (1985), 97.

[DHCB05] W. Dür, M. Hein, J. I. Cirac, H.-J. Briegel. *Standard forms of noisy quantum operations via depolarization*. Phys. Rev. A **72** (2005), 052326.

[DHH$^+$05] W. Dür, L. Hartmann, M. Hein, M. Lewenstein, H.-J. Briegel. *Entanglement in spin chains and lattices with long-range Ising-type interactions*. Phys. Rev. Lett. **94** (2005), 097203.

[DiV00] D. P. DiVincenzo. *The physical implementation of quantum computation*. Fortschr. Phys. **48** (2000), 771.

[DKSV04] A. J. Daley, C. Kollath, U. Schollwöck, G. Vidal. *Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces*. J. Stat. Mech.: Theor. Exp. (2004), P04005. ArXiv: cond-mat/0403313.

[DM03] J. Dehaene, B. De Moor. *Clifford group, stabilizer states, and linear and quadratic operations over GF(2)*. Phys. Rev. A **68** (2003), 042318.

[Dom96] C. Domb. *The critical point*. Taylor & Francis, London (1996).

[EB01] J. Eisert, H. J. Briegel. *Schmidt measure as a tool for quantifying multiparticle entanglement*. Phys. Rev. A **64** (2001), 022306.

[Eis06] J. Eisert. *Computational difficulty of global variations in the density matrix renormalization group*. Phys. Rev. Lett. **97** (2006), 260501.

[ELM93] H. G. Evertz, G. Lana, M. Marcu. *Cluster algorithm for vertex models*. Phys. Rev. Lett. **70** (1993), 875.

[EM94] R. Egger, C. H. Mak. *Low-temperature dynamical simulation of spin-boson systems*. Phys. Rev. B **50** (1994), 15210.

[EO06] J. Eisert, T. J. Osborne. *General entanglement scaling laws from time evolution*. Phys. Rev. Lett. **97** (2006), 150404.

[EP03] J. Eisert, M. B. Plenio. *Introduction to the basics of entanglement theory in continuous-variable systems*. Int. J. Quant. Inf. **1** (2003), 479.

[EPR35] A. Einstein, B. Podolsky, N. Rosen. *Can quantum-mechanical description of physical reality be considered complete?* Phys. Rev. **47** (1935), 777.

[FE04] S. Flammia, B. Eastin. *Q-circuit*. `http://info.phys.unm.edu/Qcircuit/` (2004).

[Fey82] R. P. Feynman. *Simulating physics with computers*. Int. J. Theor. Phys. **21** (1982). Reprinted in: J. G. Hey (Ed.): *Feynman and computation*, Perseus, Reading (1998).

[Fey85] R. P. Feynman. *Quantum mechanical computers*. Optics News **11** (1985), Feb. issue, 11. Reprinted in [Fey96] and in Found. Phys. **16** (1986), 507–531.

[Fey96] R. P. Feynman. *Feynman lectures on computation*. Westview Press, Boulder and Oxford (1996).

[Fis98] M. E. Fisher. *Renormalization group theory: Its basis and formulation in statistical physics*. Rev. Mod. Phys. **70** (1998), 653.

[FKK06] Y. Fujihara, A. Koga, N. Kawakami. *Variational Monte Carlo study of Hubbard model with harmonic confinement*. J. Magn. Mag. Mat. **310** (2006), 882.

[FNW92] M. Fannes, B. Nachtergaele, R. F. Werner. *Finitely correlated states on quantum spin chains*. Commun. Math. Phys. **144** (1992), 443.

[FSF] The Free Software Foundation. *The GNU project*. `http://www.gnu.org/`.

[FWGF89] M. P. A. Fisher, P. B. Weichman, G. Grinstein, D. S. Fisher. *Boson localization and the superfluid-insulator transition*. Phys. Rev. B **40** (1989), 546.

[GC99] D. Gottesman, I. L. Chuang. *Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations*. Nature **402** (1999), 390.

[GCC] The GCC Team. *The GNU Compiler Collection*. Software at `http://gcc.gnu.org`.

[GKR02] M. Grassl, A. Klappenecker, M. Rötteler. *Graphs, quadratic forms, and quantum codes*. In: *Proc. 2002 IEEE Int. Symp. on Information Theory (ISIT)*, p. 45. IEEE (2002). ArXiv: quant-ph/0703112.

[GME$^+$02] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, I. Bloch. *Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms*. Nature **415** (2002), 39.

[GN07] D. Gross, M. Van den Nest. *The LU-LC conjecture, diagonal local operations and quadratic forms over GF(2)*. ArXiv: 0707.4000 [quant-ph] (2007).

[Got96] D. Gottesman. *Class of quantum error-correcting codes saturating the quantum Hamming bound*. Phys. Rev. A **54** (1996), 1862.

[Got97] D. Gottesman. *Stabilizer codes and quantum error correction*. Ph. D. thesis, California Institute of Technology, Pasadena (1997). ArXiv: quant-ph/9705052.

[Got98a] D. Gottesman. *The Heisenberg representation of quantum computers*. ArXiv: quant-ph/9807006 (1998).

[Got98b] D. Gottesman. *Theory of fault-tolerant quantum computation*. Phys. Rev. A **57** (1998), 127.

[Grä97] G. Grätzer. *Math into LaTeX*. 3rd edn. Birkhäuser, Boston and Springer, New York (1997).

[Gru99] J. Gruska. *Quantum computing*. McGraw-Hill, London (1999).

[GTJ$^+$03] M. Galassi, J. Theiler, G. Jungman, B. Gough, J. Davies, R. Priedhorsky, M. Booth, F. Rossi. *The GNU Scientific Library (GSL), version 1.3*. `http://www.gnu.org/software/gsl/` (2003). Software.

[Gut63] M. C. Gutzwiller. *Effect of correlation on the ferromagnetism of transition metals*. Phys. Rev. Lett. **10** (1963), 159.

[Has70] W. K. Hastings. *Monte Carlo sampling methods using Markov chains and their applications*. Biometrika **57** (1970), 97.

[HBB99] M. Hillery, V. Bužek, A. Berthiaume. *Quantum secret sharing*. Phys. Rev. A **59** (1999), 1829.

[HCDB05] L. Hartmann, J. Calsamiglia, W. Dür, H.-J. Briegel. *Spin gases as microscopic models for non-Markovian decoherence*. Phys. Rev. A **72** (2005), 052107.

[HDB05] M. Hein, W. Dür, H.-J. Briegel. *Entanglement properties of multipartite entangled states under the influence of decoherence*. Phys. Rev. A **71** (2005), 032350.

[HDE$^+$06] M. Hein, W. Dür, J. Eisert, R. Raußendorf, M. Van den Nest, H.-J. Briegel. *Entanglement in graph states and its applications*. In: G. Casati, D. Shepelyansky, P. Zoller, G. Benenti (Eds.), *Quantum computers, algorithms and chaos*. *International School of Physics Enrico Fermi*, vol. 162. IOS Press, Amsterdam (2006). ArXiv: quant-ph/0602096.

[HEB04] M. Hein, J. Eisert, H. J. Briegel. *Multi-party entanglement in graph states*. Phys. Rev. A **69** (2004), 062311.

[Hen84] M. Henkel. *Statistical mechanics of the 2D quantum XY model in a transverse field*. J. Phys. A: Math. Gen. **17** (1984), L795.

[HH64] J. M. Hammersley, D. S. Handscomb. *Monte Carlo methods*. J. Wiley & Sons Inc., New York (1964).

[HHHH07] R. Horodecki, P. Horodecki, M. Horodecki, K. Horodecki. *Quantum entanglement*. ArXiv: quant-ph/0702225v2 (2007).

[HHO90] H.-X. He, C. J. Hamer, J. Oitmaa. *High-temperature series expansions for the (2+1)-dimensional Ising model*. J. Phys. A: Math. Gen. **23** (1990), 1775.

[IJK05] A. R. Its, B.-Q. Jin, V. E. Korepin. *Entanglement in the XY spin chain*. J. Phys. A: Math. Gen. **38** (2005), 2975.

[Ino00] K. Inoue. *Functional dendrimers, hyperbranched and star polymers*. Prog. Polym. Sci. **25** (2000), 453.

[Jas55] R. Jastrow. *Many-body problem with strong forces*. Phys. Rev. **98** (1955), 1479.

[JBC$^+$98] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, P. Zoller. *Cold bosonic atoms in optical lattices*. Phys. Rev. Lett. **81** (1998), 3108.

[JK04] B.-Q. Jin, V. E. Korepin. *Quantum spin chain, Toeplitz determinants and the Fisher–Hartwig conjecture*. J. Stat. Phys. **116** (2004), 79.

[JL03] R. Jozsa, N. Linden. *On the role of entanglement in quantum-computational speed-up*. Proc. Roy. Soc. A **459** (2003), 2011.

[Jos99] N. M. Josuttis. *The C++ standard library: a tutorial and reference*. Addison-Wesley, Boston etc. (1999).

[Kad00] L. P. Kadanoff. *Statistical physics: Statics, dynamics and renormalization*. World Scientific, Singapore etc. (2000).

[KADB06] C. Kruszynska, S. Anders, W. Dür, H. J. Briegel. *Quantum communication cost of preparing multipartite entanglement*. Phys. Rev. A **73** (2006), 062328.

[Kan98] B. E. Kane. *A silicon-based nuclear spin quantum computer*. Nature **393** (1998), 133.

[Kat62] S. Katsura. *Statistical mechanics of the anisotropic linear Heisenberg model*. Phys. Rev. **127** (1962), 1508.

[KGV83] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. *Optimization by simulated annealing*. Science **220** (1983), 671.

[KH04] N. Kawashima, K. Harada. *Recent developments of world-line Monte Carlo methods*. J. Phys. Soc. Jpn. **73** (2004), 1379.

[Kit97] A. Y. Kitaev. *Quantum error correction with imperfect gates*. In: A. S. Holevo, O. Hirota, C. M. Caves (Eds.), *Quantum communication, computing and measurement (Proc. 3rd Int. Conf. of Quantum Communication and Measurement)*, pp. 181–188. Plenum Press, New York (1997).

[Kit03] A. Y. Kitaev. *Fault-tolerant quantum computation by anyons*. Annals of Physics **303** (2003), 2. ArXiv: quant-ph/9707021 (1997). Note that the preprint precedes the journal publication by 6 years.

[KL96] E. Knill, R. Laflamme. *Concatenated quantum codes*. ArXiv: quant-ph/9608012 (1996).

[KLM01] E. Knill, R. Laflamme, G. J. Milburn. *A scheme for efficient quantum computation with linear optics*. Nature **409** (2001), 46.

[KLZ96] E. Knill, R. Laflamme, W. Zurek. *Threshold accuracy for quantum computation*. ArXiv: quant-ph/9610011 (1996).

[KLZ98a] E. Knill, R. Laflamme, W. H. Zurek. *Resilient quantum computation*. Science **279** (1998), 342.

[KLZ98b] E. Knill, R. Laflamme, W. H. Zurek. *Resilient quantum computation: error models and thresholds*. Proc. Roy. Soc. Lon. A **454** (1998), 365 .

[KM05] J. P. Keating, F. Mezzadri. *Entanglement in quantum spin chains, symmetry classes of random matrices, and conformal field theory*. Phys. Rev. Lett. **94** (2005), 050501.

[KMW02] D. Kielpinski, C. Monroe, D. J. Wineland. *Architecture for a large-scale ion-trap quantum computer*. Nature **417** (2002), 709.

[Kni05] E. Knill. *Quantum computing with realistically noisy devices*. Nature **434** (2005), 39.

[Kol83] M. Kolb. *Monte Carlo renormalization group for quantum systems*. Phys. Rev. Lett. **51** (1983), 1696.

[Kor04] V. E. Korepin. *Universality of entropy scaling in one dimensional gapless models*. Phys. Rev. Lett. **92** (2004), 096402.

[LA04] M. Lutz, D. Ascher. *Learning Python*. 2nd edn. O'Reilly, Sebastopol etc. (2004).

[Lan50] C. Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. J. Res. Nat. Bureau of Standards **45** (1950), 255.

[Lat07] J. I. Latorre. *Entanglement entropy and the simulation of quantum mechanics*. J. Phys. A: Math. Theor. **40** (2007), 6689.

[LB05] D. P. Landau, K. Binder. *A guide to Monte Carlo simulations in statistical physics*. 2nd edn. Cambridge University Press (2005).

[LD98] D. Loss, D. P. DiVincenzo. *Quantum computation with quantum dots*. Phys. Rev. A **57** (1998), 120.

[LDA88] S. Liang, B. Doucot, P. W. Anderson. *Some new variational resonating-valence-bond-type wave functions for the spin-½ antiferromagnetic Heisenberg model on a square lattice*. Phys. Rev. Lett. **61** (1988), 365.

[Llo96] S. Lloyd. *Universal quantum simulators*. Science **273** (1996), 1073.

[LLRV05] J. I. Latorre, C. A. Lutken, E. Rico, G. Vidal. *Fine-grained entanglement loss along renormalization-group flows*. Phys. Rev. A **71** (2005), 034301.

[LMSY96] R. Lehoucq, K. Maschhoff, D. Sorensen, C. Yang. *ARPACK*. `http://www.caam.rice.edu/software/ARPACK/` (1996).

[LRV04] J. I. Latorre, E. Rico, G. Vidal. *Ground state entanglement in quantum spin chains*. Quant. Inf. Comp. **4** (2004), 48.

[LSA+07] M. Lewenstein, A. Sanpera, V. Ahufinger, B. Damski, A. Sen De, U. Sen. *Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond*. Adv. Phys. **56** (2007), 243. ArXiv: cond-mat/0606771.

[LXW03] H. G. Luo, T. Xiang, X. Q. Wang. *Comment on "Time-dependent density-matrix renormalization group: a systematic method for the study of quantum many-body out-of-equilibrium systems"*. Phys. Rev. Lett. **91** (2003), 049701.

[Man06] S. R. Manmana. *Nonequilibrium dynamics of strongly correlated quantum systems*. Ph. D. thesis, Universität Stuttgart (2006). [web link].

[Mar03] A. Martelli. *Python in a nutshell*. O'Reilly, Sebastopol etc. (2003).

[McM65] W. L. McMillan. *Ground state of liquid He^4*. Phys. Rev. **138** (1965), A442.

[MG04] F. Mittelbach, M. Goossens. *The LaTeX companion*. 2nd edn. Addison-Wesley, Boston etc. (2004).

[MMN05] S. R. Manmana, A. Muramatsu, R. M. Noack. *Time evolution of one-dimensional quantum many body systems*. In: A. Avella, F. Mancini (Eds.), *Lectures on the physics of highly correlated electron systems IX: 9th training course in the physics of correlated electron systems and high-T_c superconductors*. *AIP Conf. Proc.*, vol. 789, pp. 269–278. American Institute of Physics (2005). [web link].

[MOL+99] J. E. Mooij, T. P. Orlando, L. Levitov, L. Tian, C. H. van der Wal, S. Lloyd. *Josephson persistent-current qubit*. Science **285** (1999), 1036.

[MPP+98] M. Murao, M. B. Plenio, S. Popescu, V. Vedral, P. L. Knight. *Multiparticle entanglement purification protocols*. Phys. Rev. A **57** (1998), R4075.

[MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller. *Equation of state calculations by fast computing machines*. J. Chem. Phys. **21** (1953), 1087.

[MRS02] M. A. Martín-Delgado, J. Rodriguez-Laguna, G. Sierra. *Density-matrix renormalization-group study of excitons in dendrimers*. Phys. Rev. B **65** (2002), 155116.

[MS73] C. B. Moler, G. W. Stewart. *An algorithm for generalized matrix eigenvalue problems*. SIAM J. Numer. Anal. **10** (1973), 241.

[MS02] E. N. Maneva, J. A. Smolin. *Improved two-party and multi-party purification protocols.* In: J. Samuel, J. Lomonaco (Eds.), *Quantum Computation and Quantum Information. AMS Contemporary Mathematics*, vol. 305. American Mathematical Society, Providence (2002). Also ArXiv: quant-ph/0003099.

[MS07] I. Markov, Y. Shi. *Simulating quantum computation by contracting tensor networks*. ArXiv: quant-ph/0511069v4 (2007). To appear in SIAM J. Comp.

[MSS99] Y. Makhlin, G. Scöhn, A. Shnirman. *Josephson-junction qubits with controlled couplings*. Nature **398** (1999), 305.

[MT94] J. J. Moré, D. J. Thuente. *Line search algorithms with guaranteed sufficient decrease*. ACM Trans. Math. Softw. (1994), 286.

[MVC07] V. Murg, F. Verstraete, J. I. Cirac. *Variational study of hard-core bosons in a two-dimensional optical lattice using projected entangled pair states*. Phys. Rev. A **75** (2007), 033605.

[MW94] C. J. Morningstar, M. Weinstein. *Contractor renormalization group method: A new computational technique for lattice systems*. Phys. Rev. Lett. **73** (1994), 1873.

[MW96] C. J. Morningstar, M. Weinstein. *Contractor renormalization group technology and exact Hamiltonian real-space renormalization group transformations*. Phys. Rev. D **54** (1996), 4131.

[NC00] M. A. Nielsen, I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, UK (2000).

[NDM04a] M. Van den Nest, J. Dehaene, B. De Moor. *Efficient algorithm to recognize the local Clifford equivalence of graph states*. Phys. Rev. A **70** (2004), 034302.

[NDM04b] M. Van den Nest, J. Dehaene, B. De Moor. *Graphical description of the action of local Clifford transformations on graph states*. Phys. Rev. A **69** (2004), 022316.

[NDM05a] M. Van den Nest, J. Dehaene, B. De Moor. *Finite set of invariants to characterize local Clifford equivalence of stabilizer states*. Phys. Rev. A **72** (2005), 014307.

[NDM05b] M. Van den Nest, J. Dehaene, B. De Moor. *Local unitary versus local Clifford equivalence of stabilizer states*. Phys. Rev. A **71** (2005), 062323.

[NH95] I. Najfeld, T. F. Havel. *Derivatives of the matrix exponential and their computation*. Advanc. Appl. Math. **16** (1995), 321.

[Nis95] T. Nishino. *Density matrix renormalization group method for 2D classical models*. J. Phys. Soc. Jap. **64** (1995), 3598.

[NM64] J. A. Nelder, R. Mead. *A simplex method for function minimization*. Computer J. **7** (1964), 308.

[NMDB06] M. Van den Nest, A. Miyake, W. Dür, H. J. Briegel. *Universal resources for measurement-based quantum computation*. Phys. Rev. Lett. **97** (2006), 150504.

[NU99] M. P. Nightingale, C. J. Umrigar (Eds.). *Quantum Monte Carlo methods in physics and chemistry*, *NATO Science Series*, vol. C 525. Kluwer, Dordrecht etc. (1999).

[Num] *NumPy*. `http://numpy.scipy.org/`.

[NW99] J. Nocedal, S. J. Wright. *Numerical optimization*. Springer (1999).

[OAFF02] A. Osterloh, L. Amico, G. Falci, R. Fazio. *Scaling of entanglement close to a quantum phase transition*. Nature **416** (2002), 608.

[Oli06] T. E. Oliphant. *Guide to NumPy*. Trelgol, `http://www.trelgol.com` (2006).

[ON02] T. J. Osborne, M. A. Nielsen. *Entanglement, quantum phase transitions, and density matrix renormalization*. Quant. Inf. Proc. **1** (2002), 45.

[Ons44] L. Onsager. *Crystal statistics. I. A two-dimensional model with an order-disorder transition*. Phys. Rev. **65** (1944), 117.

[OP05] M. Ostilli, C. Presilla. *Exact Monte Carlo time dynamics in many-body lattice quantum systems*. J. Phys. A: Math. Gen. **38** (2005), 405.

[ÖR95] S. Östlund, S. Rommer. *Thermodynamic limit of density matrix renormalization*. Phys. Rev. Lett. **75** (1995), 3537.

[Pak02] S. Pakin. *The comprehensive LaTeX symbol list*. On the Comprehensive TeX Archive Network (`http://www.ctan.org`) (2002).

[Pan97] T. Pang. *An introduction to computational physics*. Cambridge University Press, Cambridge, UK (1997).

[Pap94] C. Papadimitriou. *Computational complexity*. Addison Wesley, Reading (1994).

[PEDC05] M. B. Plenio, J. Eisert, J. Dreissig, M. Cramer. *Entropy, entanglement, and area: Analytical results for harmonic lattice systems*. Phys. Rev. Lett. **94** (2005), 060503.

[Pes04] I. Peschel. *On the entanglement entropy for an XY spin chain*. J. Stat. Mech.: Th. Exp. (2004), P12005.

[Pet] P. Peterson. *F2PY: Fortran to Python interface generator*. `http://cens.ioc.ee/projects/f2py2e/`.

[Pfe69] P. Pfeuty. *The one-dimensional Ising model with a transverse field*. Annals of Physics **57** (1969), 79.

[PlM] *Derivative of inverse matrix*. PlanetMath.Org (a web encyclopædia), `http://planetmath.org/encyclopedia/DerivativeOfInverseMatrix.html` (2006). Version 3.

[Pow64] M. J. D. Powell. *An efficient method for finding the minimum of a function of several variables without calculating derivatives*. Computer J. **7** (1964), 155 .

[PRB06] G. Pupillo, A. M. Rey, G. G. Batrouni. *Bragg spectroscopy of trapped one-dimensional strongly interacting bosons in optical lattices: Probing the cake structure*. Phys. Rev. A **74** (2006), 013601.

[Pre98a] J. Preskill. *Fault-tolerant quantum computation*. In: H.-K. Lo, S. Popescu, T. P. Spiller (Eds.), *Introduction to quantum computation*. World Scientific, Singapore etc. (1998). ArXiv: quant-ph/9712048.

[Pre98b] J. Preskill. *Lecture notes for Physics 229: Quantum information and computation*. `http://www.theory.caltech.edu/~preskill/ph229/#lecture`, California Institute of Technology (1998).

[Pre98c] J. Preskill. *Reliable quantum computers*. Proc. Roy. Soc. Lon. A **454** (1998), 385 .

[PST98] N. V. Prokof'ev, B. V. Svistunov, I. S. Tupitsyn. *Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems*. Zh. Éksp. Teor. Fiz. **114** (1998), 570. Also: J. Exp. Th. Phys. **87** (1998), 310.

[Puz05] R. Puzio. *Cayley's parameterization of orthogonal matrices*. PlanetMath.Org (a web encyclopædia), `http://planetmath.org/?op=getobj&from=objects&id=6535` (2005). Version 12.

[PWKH99] I. Peschel, X. Wang, M. Kaulke, K. Hallberg (Eds.). *Density-matrix renormalization*, *Lecture Notes in Physics*, vol. 528. Springer, Berlin etc. (1999).

[R$^+$] G. van Rossum, et al. *Python* [a programming language]. `http://www.python.org`.

[RBB03] R. Raußendorf, D. E. Browne, H.-J. Briegel. *Measurement-based quantum computation on cluster states*. Phys. Rev. A **68** (2003), 022312.

[RH07] R. Raussendorf, J. Harrington. *Fault-tolerant quantum computation with high threshold in two dimensions*. Phys. Rev. Lett. **98** (2007), 190504.

[RL85] H. De Raedt, A. Lagendijk. *Monte Carlo simulation of quantum statistical lattice models*. Phys. Rep. **127** (1985), 233.

[RL92] H. De Raedt, W. von der Linden. *Quantum lattice problems*. In: K. Binder (Ed.), *The Monte Carlo method in condensed matter physics*. *Topics in Applied Physics*, vol. 71, pp. 249–281. Springer, Berlin etc. (1992). [DOI link].

[RÖ97] S. Rommer, S. Östlund. *Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group*. Phys. Rev. B **55** (1997), 2164.

[Rod02] J. Rodríguez Laguna. *Real space renormalization group techniques and applications*. Ph. D. thesis, Universidad Complutense de Madrid (2002). ArXiv: cond-mat/0207340.

[RT87] A. H. G. Rinnooy Kan, G. T. Timmer. *Stochastic global optimization methods. Part I: Clustering methods*. Mathematical Programming **39** (1987), 27.

[Run92] K. J. Runge. *Finite-size study of the ground-state energy, susceptibility, and spin-wave velocity for the Heisenberg antiferromagnet*. Phys. Rev. B **45** (1992), 12292.

[Sac99] S. Sachdev. *Quantum phase transitions*. Cambridge Univesity Press, Cambridge (1999).

[Sch] J. Schulenburg. *Spinpack*. `http://www-e.uni-magdeburg.de/jschulen/spin/index.html`. Software, 2003–2007.

[Sch35] E. Schrödinger. *Die gegenwärtige Situation in der Quantenmechanik*. Naturwissenschaften **23** (1935), 807. Reprinted in E. Schrödinger: *Gesammelte Abhandlungen*, vol. 3. Verlag der Österreichischen Akademie der Wissenschaften, Wien (1984).

[Sch01] D. Schlingemann. *Stabilizer codes can be realized as graph codes*. ArXiv: quant-ph/0111080 (2001).

[Sch02] F. Schoen. *Two-phase methods for global optimization*. In: P. M. Pardalos, H. E. Romeijn (Eds.), *Handbook of global optimization, Vol. 2*. Kluwer (2002). ArXiv: quant-ph/0602096.

[Sch05] U. Schollwöck. *The density-matrix renormalization group*. Rev. Mod. Phys. **77** (2005), 259.

[SDV06] Y.-Y. Shi, L.-M. Duan, G. Vidal. *Classical simulation of quantum many-body systems with a tree tensor network*. Phys. Rev. A **74** (2006), 022320.

[Sed88] R. Sedgewick. *Algorithms*. 2nd edn. Addison-Wesley, Boston etc. (1988).

[Sho94] P. W. Shor. *Algorithms for quantum computation: discrete logarithms and factoring*. In: *35th Annual Symp. on Foundations of Computer Science*, pp. 124–134. IEEE (1994).

[Sho95] P. W. Shor. *Scheme for reducing decoherence in quantum computer memory*. Phys. Rev. A **52** (1995), R2493.

[Sho96] P. W. Shor. *Fault-tolerant quantum computation*. In: *Proc. 37th Symp. on Foundations of Computing*, pp. 56–65. IEEE Computer Society Press (1996). ArXiv: quant-ph/9605011.

[Sib81] R. Sibson. *A brief description of natural neighbor interpolation*. In: V. Barnett (Ed.), *Interpreting multivariate data*, pp. 21–36. John Wiley & Sons, New York (1981).

[SMK77] M. Suzuki, S. Miyashita, A. Kuroda. *Monte Carlo simulation of quantum spin systems. I*. Prog. Theo. Phys. **58** (1977), 1377.

[SP97] R. Storn, K. Price. *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*. J. Global Optimization **11** (1997), 341.

[SS02] P. J. Salas, A. L. Sanz. *Numerical simulation of information recovery in quantum computers*. Phys. Rev. A **66** (2002), 022302.

[SS04] P. J. Salas, A. L. Sanz. *Effect of ancilla's structure on quantum error correction using the seven-qubit Calderbank-Shor-Steane code*. Phys. Rev. A **69** (2004), 052322.

[STD05] K. M. Svore, B. M. Terhal, D. P. DiVincenzo. *Local fault-tolerant quantum computation*. Phys. Rev. A **72** (2005), 022317.

[Ste72] G. W. Stewart. *On the sensitivity of the eigenvalue problem* $Ax = \lambda Bx$. SIAM J. Numer. Anal. **9** (1972), 669.

[Ste96a] A. Steane. *Multiple-particle interference and quantum error correction*. Proc. Roy. Soc. A **452** (1996), 2551.

[Ste96b] A. M. Steane. *Error correcting codes in quantum theory*. Phys. Rev. Lett. **77** (1996), 793.

[Ste98a] A. Steane. *Space, time, parallelism and noise requirements for reliable quantum computing*. Fortschr. Phys. **46** (1998), 443.

[Ste98b] A. M. Steane. *Efficient fault-tolerant quantum computing*. ArXiv: quant-ph/9809054 (1998).

[Ste02] A. M. Steane. *Quantum computer architecture for fast entropy extraction*. Quant. Inf. Comp. **2** (2002), 297. ArXiv: quant-ph/0203047.

[Ste03] A. M. Steane. *Overhead and noise threshold of fault-tolerant quantum error correction*. Phys. Rev. A **68** (2003), 042322.

[Str97] B. Stroustrup. *The C++ programming language*. 3rd edn. Addison-Wesley, Boston etc. (1997).

[Suz76] M. Suzuki. *Relationship between d-dimensional quantal spin systems and (d+1)-dimensional Ising systems — Equivalence, critical exponents and systematic approximants of the partition function and spin correlations*. Prog. Theo. Phys. **54** (1976), 1454.

[SV07] A. W. Sandvik, G. Vidal. *Variational quantum Monte Carlo simulations with tensor-network states*. ArXiv: 0708.2232 [cond-mat] (2007).

[SW87] R. H. Swendsen, J.-S. Wang. *Nonuniversal critical dynamics in Monte Carlo simulations*. Phys. Rev. Lett. **58** (1987), 86.

[SW02] D. Schlingemann, R. F. Werner. *Quantum error-correcting codes associated with graphs*. Phys. Rev. A **65** (2002), 012308.

[SW06] M. S. Siu, M. Weinstein. *Exploring contractor renormalization: New perspectives and applications to the 2-D Heisenberg antiferromagnet*. ArXiv: cond-mat/0608042 (2006).

[Swe79] R. H. Swendsen. *Monte Carlo renormalization group*. Phys. Rev. Lett. **42** (1979), 859.

[SWVC07a] N. Schuch, M. M. Wolf, F. Verstraete, J. I. Cirac. *Entropy scaling and simulability by matrix product states*. ArXiv: 0705.0292 [quant-ph] (2007).

[SWVC07b] N. Schuch, M. M. Wolf, F. Verstraete, J. I. Cirac. *String states: Properties and applications*. ArXiv: 0708.1567 [quant-ph] (2007).

[TAH98] M. Troyer, B. Ammon, E. Heeb. *Parallel object oriented Monte Carlo simulations*. In: D. Caromel, R. R. Oldehoeft, M. Tholburn (Eds.), *Computing in object-oriented parallel environments (Proc. ISCOPE 1998)*. *Lecture Notes in Computer Science*, vol. 1505, p. 191. Springer, Berlin etc. (1998). [web link].

[Tak99] M. Takahashi. *Thermodynamics of one-dimensional solvable models*. Cambridge University Press, Cambridge, UK (1999).

[TB97] L. N. Trefethen, D. Bau. *Numerical linear algebra*. SIAM, Philadelphia (1997).

[TC89] N. Trivedi, D. M. Ceperley. *Green-function Monte Carlo study of quantum antiferromagnets*. Phys. Rev. B **40** (1989), 2737.

[TED$^+$05] J. M. Taylor, H.-A. Engel, W. Dür, A. Yacoby, C. M. Marcus, P. Zoller, M. D. Lukin. *Fault-tolerant architecture for quantum computation using electrically controlled semiconductor spins*. Nature Phys. **1** (2005), 177.

[Tof80] T. Toffoli. *Reversible computing*. In: J. W. de Bakker, J. van Leeuwen (Eds.), *Automata, languages, and programming*. *Lecture Notes in Computer Sciences*, vol. 85, pp. 632–644. Springer, Berlin etc. (1980). [DOI link].

[TW05] M. Troyer, U.-J. Wiese. *Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations*. Phys. Rev. Lett. **94** (2005), 170201.

[Unr95] W. G. Unruh. *Maintaining coherence in quantum computers*. Phys. Rev. A **51** (1995), 992.

[VC04a] F. Verstraete, J. I. Cirac. *Renormalization algorithms for quantum-many body systems in two and higher dimensions*. ArXiv: cond-mat/0407066 (2004).

[VC04b] F. Verstraete, J. I. Cirac. *Valence-bond states for quantum computation*. Phys. Rev. A **70** (2004), 060302.

[VGC04] F. Verstraete, J. J. García-Ripoll, J. I. Cirac. *Matrix product density operators: simulation of finite-temperature and dissipative systems*. Phys. Rev. Lett. **93** (2004), 207204.

[Vid03] G. Vidal. *Efficient classical simulation of slightly entangled quantum computations*. Phys. Rev. Lett. **91** (2003), 147902.

[Vid04] G. Vidal. *Efficient simulation of one-dimensional quantum many-body systems*. Phys. Rev. Lett. **93** (2004), 040502.

[Vid05] G. Vidal. *Entanglement renormalization*. ArXiv: cond-mat/0512165 (2005).

[VLRK03] G. Vidal, J. I. Latorre, E. Rico, A. Kitaev. *Entanglement in quantum critical phenomena*. Phys. Rev. Lett. **90** (2003), 227902.

[Voj03] M. Vojta. *Quantum phase transitions*. Rep. Prog. Phys. **66** (2003), 2069.

[Vol84] D. Vollhardt. *Normal ^{3}He: an almost localized Fermi liquid*. Rev. Mod. Phys. **56** (1984), 99.

[VPC04] F. Verstraete, D. Porras, J. I. Cirac. *Density matrix renormalization group and periodic boundary conditions: a quantum information perspective*. Phys. Rev. Lett. **93** (2004), 227205.

[VWPC06] F. Verstraete, M. M. Wolf, D. Perez-Garcia, J. I. Cirac. *Criticality, the area law, and the computational power of projected entangled pair states*. Phys. Rev. Lett. **96** (2006), 220601.

[WATB04] S. Wessel, F. Alet, M. Troyer, G. G. Batrouni. *Quantum Monte Carlo simulations of confined bosonic atoms in optical lattices*. Phys. Rev. A **70** (2004), 053615.

[WBI$^+$92] D. J. Wineland, J. J. Bollinger, W. M. Itano, F. L. Moore, D. J. Heinzen. *Spin squeezing and reduced quantum noise in spectroscopy*. Phys. Rev. A **46** (1992), R6797.

[WBIH94] D. J. Wineland, J. J. Bollinger, W. M. Itano, D. J. Heinzen. *Squeezed atomic states and projection noise in spectroscopy*. Phys. Rev. A **50** (1994), 67.

[Wei05] E. W. Weisstein. *Buffon's needle problem*. In: MathWorld – A Wolfram Web Resource, `http://mathworld.wolfram.com/BuffonsNeedleProblem.html` (2005).

[Wer] R. F. Werner (Ed.). *Open problems in quantum information theory*. `http://www.imaph.tu-bs.de/qi/problems/`.

[Weß05] S. Weßel. *Quantum Monte Carlo studies of strongly correlated systems*. Habilitationsschrift, Universität Stuttgart (2005). [web link].

[WF04] S. R. White, A. E. Feiguin. *Real-time evolution using the density matrix renormalization group*. Phys. Rev. Lett. **93** (2004), 076401.

[Whi92] S. R. White. *Density matrix formulation for quantum renormalization groups*. Phys. Rev. Lett. **69** (1992), 2863.

[Whi93] S. R. White. *Density-matrix algorithms for quantum renormalization groups*. Phys. Rev. B **48** (1993), 10345.

[Whi98] S. R. White. *Strongly correlated electron systems and the density renormalization group*. Phys. Rep. **301** (1998), 187.

[Wil75] K. G. Wilson. *The renormalization group: Critical phenomena and the Kondo problem*. Rev. Mod. Phys. **47** (1975), 773.

[Wil93] K. G. Wilson. *The renormalization group and critical phenomena*. In: G. Ekspong (Ed.), *Nobel lectures in physics 1981 – 1990*. World Scientific, Singapore (1993). Lecture given 1982, [web link].

[WN92] S. R. White, R. M. Noack. *Real-space quantum renormalization groups*. Phys. Rev. Lett. **68** (1992), 3487.

[WOH94] Z. Weihong, J. Oitmaa, C. J. Hamer. *Series expansions for the 3D transverse Ising model at T=0*. J. Phys. A: Math. Gen. **27** (1994), 5425.

[Wol06] M. M. Wolf. *Violation of the entropic area law for fermions*. Phys. Rev. Lett. **96** (2006), 010404.

[WPD01] R. C. Whaley, A. Petitet, J. J. Dongarra. *Automated empirical optimizations of software and the ATLAS project*. Parallel Computing **27** (2001), 3.

[WS99] D. J. Wales, H. A. Scheraga. *Global optimization of clusters, crystals, and biomolecules*. Science **285** (1999), 1368.

[WVHC07] M. M. Wolf, F. Verstraete, M. B. Hastings, J. I. Cirac. *Area laws in quantum systems: mutual information and correlations*. ArXiv: 0704.3906 [quant-ph] (2007).

[WZ82] W. K. Wootters, W. H. Zurek. *A single quantum cannot be cloned*. Nature **299** (1982), 802.

[YS87] H. Yokoyama, H. Shiba. *Variational Monte-Carlo studies of Hubbard model. I*. J. Phys. Soc. Jpn. **56** (1987), 1490.

[Zal96] C. Zalka. *Threshold estimate for fault tolerant quantum computation*. ArXiv: quant-ph/9612028 (1996).

[ZBLN97] C. Zhu, R. H. Byrd, P. Lu, J. Nocedal. *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*. ACM Trans. Math. Softw. **23** (1997), 550.

[ZCC07] B. Zeng, A. Cross, I. L. Chuang. *Transversality versus universality for additive quantum codes*. ArXiv: 0706.1382 [quant-ph] (2007).

[ZCCC07] B. Zeng, H. Chung, A. W. Cross, I. L. Chuang. *Local unitary versus local Clifford equivalence of stabilizer and graph states*. Phys. Rev. A **75** (2007), 032325.

[ZLC00] X. Zhou, D. W. Leung, I. L. Chuang. *Methodology for quantum logic gate construction*. Phys. Rev. A **62** (2000), 052316.

[Zur03] W. H. Zurek. *Decoherence, einselection, and the quantum origins of the classical*. Rev. Mod. Phys. **75** (2003), 715.

[ZV04] M. Zwolak, G. Vidal. *Mixed-state dynamics in one-dimensional quantum lattice systems: a time-dependent superoperator renormalization algorithm*. Phys. Rev. Lett. **93** (2004), 207205.

Appendix C

Curriculum vitae

22 Dec 1976	born in Altenerding, Bavaria, Germany Parents: Edith Anders, née Orawa, and Dr. Rainer Anders grown up in Munich, Germany
1996	Abitur (high school exam) at the Wilhelm-Hausenstein-Gymnasium, Munich
8/96 – 9/97	Zivildienst (civil service) at the Psychiatric Hospital of the Ludwig-Maximilians-Universität, Munich
7/97 – 11/02	study of physics at the Technische Universität München (Munich, Germany); finished with the degree "Diplom-Physiker" (equivalent to M. Sc.)
11/01 – 11/02	Diplomarbeit (equiv. to an M. Sc. thesis) in the group of Prof. Dirk Bouwmeester at the University of Oxford, UK, and the University of California at Santa Barbara (USA) in experimental quantum optics
04/03 – 10/07	Scientist in the group of Prof. Hans J. Briegel, working on the present Ph. D. thesis, at the Ludwig-Maximilians-Universität in Munich (until 12/03) and the University of Innsbruck, Austria